Assembler 6502

Bob Bright

Assembler 6502

Bob Bright

© Copyright per l'edizione originale: Bob Bright 1983
Edizione originale: Pitman Publishing LTD
Titolo originale: Pocket Guide ASSEMBLY LANGUAGE FOR THE 6502

© Copyright per l'edizione italiana:
GRUPPO EDITORIALE JACKSON - Settembre 1985

TRADUZIONE: Enrico Bottinelli
SUPERVISIONE TECNICA: Vittorio Riva
COPERTINA: Silvana Corbelli
FOTOCOMPOSIZIONE: Cencograf-Rotografica srl
P.zza S. Marco 1 - Milano - Tel. 655.20.13 - 655.51.45
STAMPA: Grafika '78 - Pioltello (MI)

Tutti i diritti sono riservati. Stampato in Italia. Nessuna parte di questo libro può essere riprodotta, memorizzata in sistemi di archivio, o trasmessa in qualsiasi forma o mezzo, elettronico, meccanico, fotocopia, registrazione o altri senza la preventiva autorizzazione scritta dell'editore.

INDICE ANALITICO

Accumulatore 3
Adattatore per interfaccia versatile (VIA) 6522 55
ADC 16
AND 17
ASL 18
BCC 18
BCS 19
BEQ 19
BIT 20
BMI 20
BNE 21
BPL 21
BRK 22-72
BVC 22
BVS 23
Caricamento dello stack pointer 50
CLC 23
CLD 24
CLI 24
CLV 25
CMP 25
Codice macchina 5
Configurazione di più VIA 80
Contatore di programma 2
Conteggio di impulsi con il timer 2 78
Controllo di interuttori 82
Conversione analogica/digitale 79
CPU 6502 (Unità Centrale di Elaborazione) 1
CPX 26
CPY 27
DEC 27
DEX 28
DEY 28
Emissione dalla porta B 81
EOR 29
Funzionamento dello stack 49
Generazione di un'onda quadra su PB7 con il timer 1 77
Generazione di una forma d'onda con CB2 79
Gestione degli interrupt 67
Gestione di un IRQ 76
Immissione/Emissione 54
INC 30
Indirizzamento assoluto 7
Indirizzamento assoluto indicizzato 9
Indirizzamento immediato 8
Indirizzamento implicito 12
Indirizzamento in pagina zero 8
Indirizzamento indicizzato in pagina zero 9
Indirizzamento indicizzato indiretto 10
Indirizzamento indiretto 10
Indirizzamento indiretto indicizzato 11
Indirizzamento relativo 12
Interrupt 53
Interrupt e reset 68
Interrupt non mascherabile 70
INX 30
INY 31
JMP 31
JSR 31
LDA 32
LDX 33
LDY 33
Lettura e scrittura dei registri di memoria di T1 64
Lettura e scrittura del contatore T1 63

Lettura e scrittura del contatore T2 65
Linea di controllo CA1 59
Linea di controllo CB1 e CB2 60
Linguaggio assembly 5
Listato assembler 7
LSR 34
Memorizzazione dati 61
Modello per la programmazione della CPU 2
Modifica dello stack 54
Moltiplicazione a 8 bit 75
NOP 34
Operandi 5
Operazioni di push e pull 51
ORA 35
Paginazione 14
PHA 36
PHP 36
PLA 36
PLP 37
Porte di ingresso/uscita 58
Programma sorgente 6
Registri di direzione DDRB e DDRA 58
Registri di ingresso IRB e IRA 58
Registri di uscita ORB e ORA 58
Registri indice X e Y 3
Registri interni 57
Registro degli indicatori di interrupt (registro $D) 67
Registro di abilitazione degli interrupt 68
Registro di controllo ausiliario 61
Registro di controllo periferico 59
Registro di stato 3
Reset 71
Richiesta di interrupt (IRQ) 69
Richiesta multiple di IRQ 71
ROL 37
ROR 38
RTS 39
Salvataggio dello stato della CPU 70
SBC 40
SEC 41
SED 41
SEI 42
Shift register (registro a scorrimento) 65
Somma binaria a 16 bit 74
Somma binaria a 8 bit 72
Sottrazione binaria a 8 bit 73
STA 42
Stack pointer (SP) 2
STX 43
STY 44
Subroutine 51
Subroutine nidificate 52
TAX 44
TAY 45
Tempi di esecuzione 13
Timer 1 62
Timer 2 64
TSX 46
TXA 46
TXS 47
TYA 45

UTILIZZO DELLA GUIDA

Ogni caratteristica del linguaggio assembly del 6502 ha una sezione dedicata.
La tavola I contiene l'insieme completo delle istruzioni ed include dettagli sui codici operativi, i vari modi di indirizzamento, i tempi di esecuzione ed il numero di bytes del codice macchina. In essa è anche indicata l'influenza dell'istruzione sul registro di stato.
La tavola II è una lista dei codici operativi in ordine numerico, insieme alle istruzioni mnemoniche ed ai modi di indirizzamento. Ci sono sezioni sull'interrupt e sul funzionamento dello stack insieme ad un dettagliato studio della VIA (versatile interface adapter). Dove necessario sono forniti esempi sia nel testo che alla fine della guida.

LA CPU 6502 (UNITÀ CENTRALE DI ELABORAZIONE)

Il 6502 è un microprocessore della famiglia di dispositivi 6500. Tutti questi dispositivi sono compatibili a livello software ma non tutti hanno la stessa pedinatura.
Il 6502 è un dispositivo a 40 pin (terminali) ed ha, tra le altre, le seguenti caratteristiche:
56 istruzioni
13 modi di indirizzamento
Bus di indirizzamento a 16 bits
Bus dati parallelo a 8 bits
Frequenza di clock di 1 e 2 MHz
Gestione di interrupt (interruzioni) mascherabili e non
Gestione di uno stack di 256 bytes.
Il 6502 comprende alcuni registri interni, dettagliati nella sezione del modello di programmazione della CPU, una unità aritmetico-logica, registro e decodificatore delle istruzioni, e la logica per la gestione degli interrupt. Il manuale delle caratteristiche del costruttore può essere consultato per ogni dettaglio sulla architettura del sistema, le caratteristiche elettriche, la funzione dei terminali ed i diagrammi per la temporizzazione.

MODELLO PER LA PROGRAMMAZIONE DELLA CPU

Il modello di programmazione del 6502 è illustrato sotto:

Accumulatore	7 … 0
Registro indice X	7 … 0
Registro indice Y	7 … 0
Stack pointer SP (puntatore dello stack)	7 … 0
Contatore di programma PC	15 … 7 … 0
Registro di stato P	N V B D I Z C

N Indicatore di negativo
V Indicatore di eccedenza (overflow)
non usato
B Indicatore dell'istruzione di BREAK
D Indicatore del modo decimale
I Disabilitazione dell'interrupt
Z Indicatore di zero
C Indicatore di riporto

Contatore di programma

Il contatore di programma è un registro a 16 bit che contiene l'indirizzo della prossima istruzione, o di parte di essa, nella sequenza del programma.

Stack pointer (SP)

L'area dello stack è ristretta alla pagina 01 della memoria. Lo stack pointer mantiene l'indirizzo di una locazione di memoria in questa pagina ed indica quindi la posizione corrente dello stack.

Accumulatore

L'Accumulatore è un registro a 8 bit usato in alcune operazioni aritmetiche e logiche; il risultato di queste operazioni è depositato nell'accumulatore.

Registri indice X e Y

I registri indice sono registri a 8 bit e sono usati nei modi di indirizzamento indicizzato. I registri possono anche essere usati in alcune operazioni aritmetiche e per trasferire dati tra l'accumulatore e lo stack pointer.

Registro di stato

Il registro di stato è un registro a 8 bit che contiene 6 indicatori, un bit di disabilitazione dell'interrupt ed un bit non utilizzato. Segue la descrizione di ogni bit.

Bit 0 Indicatore di riporto (C)
L'indicatore di riporto si comporta come il nono bit in alcune operazioni aritmetiche e logiche. Particolare cura deve essere posta nell'interpretazione di questo indicatore mentre si usa l'istruzione di sottrazione. Questo indicatore deve essere azzerato prima di utilizzare operazioni di somma (ADC) e posto a 1 prima delle sottrazioni (SBC).

Bit 1 Indicatore di zero (Z)
Questo indicatore è impostato a uno se il risultato di una operazione aritmetico-logica è zero; se il risultato è diverso da zero l'indicatore viene azzerato.

Bit 2 Disabilitazione dell'interrupt (I)
Questo bit abilita o meno l'interrupt mascherabile IRQ. Se è uno IRQ è disabilitato, se è zero IRQ è abilitato. Non ha effetto su NMI (interrupt non mascherabile).

Bit 3 Funzionamento decimale (D)

Se l'indicatore di modo decimale è azzerato l'unità aritmetica della CPU esegue le operazioni aritmetiche in binario. Impostando l'indicatore a uno si selezione l'aritmetica BCD (decimale). Assicurarsi sempre che l'indicatore sia impostato nel modo richiesto.

Bit 4 Indicatore di BREAK (B)
Quando la CPU esegue un'istruzione di BREAK questo indicatore viene impostato a uno (v. istruzione BRK pag. 22)

Bit 5 non utilizzato

Bit 6 Indicatore di eccedenza (V)
Durante operazioni di aritmetica binaria con segno l'indicatore verrà impostato a 1 se il risultato eccede i 7 bit, altrimenti sarà azzerato.

Bit 7 Indicatore di negativo (N)
Questo indicatore viene impostato a uno se il bit più significativo di un'operazione è uno, altrimenti viene azzerato.

Gli indicatori possono essere impostati a uno, azzerati o non influenzati dal risultato dell'istruzione appena eseguita.
La tavola I indica l'influenza di ogni istruzione sul registro di stato.
Alcuni indicatori possono essere impostati direttamente:

CLC azzera il riporto
CLD seleziona l'aritmetica binaria
CLI azzera il bit di disabilitazione dell'interrupt
CLV azzera l'indicatore di overflow (eccedenza)
SEC imposta a uno il riporto
SED seleziona l'aritmetica decimale (BCD)
SEI imposta a uno il bit di disabilitazione dell'interrupt.

Analisi degli indicatori
Quattro indicatori, indicati con N, V Z e C, possono essere analizzati e condizionare una istruzione di salto con il loro stato. Per ogni indicatore segue la lista delle istruzioni:

N BMI e BPL
V BVS e BVC
Z BEQ e BNE
C BCS e BCC

ELEMENTI DEL LINGUAGGIO

Codice macchina

Il codice macchina, o codice oggetto, di ogni istruzione del 6502 consiste in uno, due o tre bytes. Il primo byte contiene sempre il codice operativo della particolare istruzione ed il modo di indirizzamento. Per esempio, A2 è il codice per LDX (caricamento del registro X) che utilizza il modo di indirizzamento immediato. Il secondo e terzo byte, se presenti, possono contenere i dati, l'indirizzo o uno spostamento che verrà utilizzato per formare l'indirizzo.

Linguaggio assembly

In linguaggio assembly ogni istruzione consiste in un operatore ed in un operando. L'operatore è rappresentato da un codice mnemonico di tre lettere che indica l'istruzione macchina. Se si usa un assembler il codice assembler deve essere uguale al codice assembly.
Esempi:

LDA	carica l'accumulatore
BEQ	salta per uguale

Operandi

Il formato dell'operando in linguaggio assembly indica, oltre ai dati ecc., anche il modo di indirizzamento. Segue la lista dei formati con i relativi modi di indirizzamento.

# Operando	Operando immediato
Operando	pagina zero, assoluto o relativo
Operando, X	pagina zero e assoluto indicizzato X
Operando, Y	pagina zero e assoluto indicizzato Y
(Operando, X)	indicizzato indiretto
(Operando), Y	indiretto indicizzato

(Operando)	indiretto
A	accumulatore

Se non ci sono operandi il modo di indirizzamento è implicito. Il termine operando della precedente lista può essere espresso in vari modi:

1) come numero binario, decimale, ottale o esadecimale; di solito si usa distinguere i vari sistemi numerici ma il modo esatto varia a seconda dell'assembler usato. Alcuni esempi tipici sono:

esadecimale	$ 80	o	80H
ottale	@35	o	35Q
binario	%01110001	o	01110001B
decimale	79	o	79D

2) come un simbolo che dovrà essere definito e che verrà sostituito dall'assembler con un valore numerico
3) come una semplice espressione numerica contenente gli operatori +, -, /, *
4) come un carattere ASCII indicandolo con un apice (') di prefisso

Programma sorgente

Il programma sorgente è diviso in quattro campi che contengono, rispettivamente, l'etichetta, l'operatore, l'operando ed il commento. Un tipico programma sorgente può essere:

```
        ORG $3000
        LDY #$FF;     inizio routine di ritardo
LOOP2   LDX #$FF
LOOP1   DEX
        BNE LOOP1
        DEY
        BNE LOOP2; fine della routine di ritardo
```

Il primo campo è occupato da un'etichetta ed è opzionale, è

usato soprattutto in istruzioni di salto. Permette al programmatore di indicare simbolicamente gli indirizzi che verranno poi tradotti dall'assembler. Generalmente ci sono restrizioni, dipendenti dall'assembler utilizzato, nell'utilizzo dei caratteri per le etichette. L'ultimo campo è riservato ai commenti e serve per una buona documentazione del programma; è opzionale.

Listato assembler

L'assemblatore converte il sorgente in oggetto o codice macchina; la tipica stampa dell'assemblaggio del programma precedente potrebbe essere:

Indirizzo	Codice macchina	Etichetta	Linguaggio assembly	Commenti
3000	A0 FF		LDY =$FF	; inizio ritardo
3002	A2 FF	LOOP2	LDX =$FF	
3004	CA	LOOP1	DEX	
3005	D0 FD		BNE LOOP1	
3007	88		DEY	
3008	D0 F8		BNE LOOP2	; fine ritardo

MODI DI INDIRIZZAMENTO

Indirizzamento assoluto

L'indirizzamento assoluto richiede 3 bytes di codice macchina. Il primo byte contiene il codice operativo; il secondo ed il terzo contengono rispettivamente il byte meno significativo ed il byte più significativo dell'indirizzo.

Esempio:
LDA $2055
Questa istruzione carica l'accumulatore con il contenuto della locazione di memoria $2055. Dopo l'assemblaggio questa istruzione diventa: AD 55 20

Il primo byte AD è il codice operativo e gli altri due formano l'indirizzo della locazione di memoria. Notate che l'indirizzo a 16 bit è memorizzato con il byte meno significativo che precede quello più significativo.
Un altro esempio può essere: JMP $3020
Questa istruzione trasferisce il controllo del programma alla locazione $3020. Dopo l'assemblaggio otterremo: 4C 20 30.
Con l'indirizzamento assoluto possiamo accedere ad ogni bytes della memoria indirizzabile di 64K.

Indirizzamento in pagina zero

L'indirizzamento in pagina zero richiede 2 bytes di codice macchina. Il primo byte contiene il codice operativo; il secondo un indirizzo compreso tra 00 e 255 (pagina zero della memoria). I dati su cui operare si trovano a questo indirizzo.

Esempio: LDA $55
Questa istruzione carica l'accumulatore con il contenuto della locazione di memoria $0055 che è la locazione 55 della pagina zero.
L'istruzione verrà assemblata così: A5 55
Il primo byte è il codice operativo, il secondo è l'indirizzo entro la pagina zero.

Indirizzamento immediato

L'indirizzamento immediato richiede 2 bytes di codice macchina. Il primo byte contiene il codice operativo, il secondo i dati su cui operare.

Esempio: LDA #$9A
Questa istruzione carica l'accumulatore con il numero $9A.
Il simbolo # indica il modo di indirizzamento immediato. Dopo l'assemblaggio otterremo: A9 9A
Il primo byte è il codice operativo ed il secondo byte il dato.

In questo modo di indirizzamento si dice che il dato segue immediatamente il codice operativo.

Indirizzamento assoluto indicizzato

L'indirizzamento assoluto idicizzato richiede 3 bytes di codice macchina. Il primo byte contiene il codice operativo, il modo di indirizzamento e specifica l'indirizzo indice; il secondo ed il terzo byte contengono rispettivamente il byte meno significativo e più significativo di un indirizzo al quale verrà sommato il contenuto del registro indice per ottenere l'indirizzo effettivo del dato a cui si vuole accedere.

Esempio: LDA $903A,X
Se il registro X contiene il valore $30, l'accumulatore sarà caricato con il dato contenuto nella locazione $906A ($903A+$30). L'assemblaggio ci renderà: BD 3A 90
Altro esempio: LDA $90A5,Y
Se il registro indice Y contiene il numero $ A9 l'accumulatore sarà caricato con il contenuto della locazione $914E ($90A5+$A9). Dopo l'assemblaggio avremo B9 A5 90
Note:

1) l'indirizzo a 16 bit è memorizzato con il byte meno significativo seguito da quello più significativo
2) non ci sono problemi nell'attraversare le pagine di memoria ma con alcune istruzioni il tempo di esecuzione può variare.

Indirizzamento indicizzato in pagina zero

L'indirizzamento indicizzato in pagina zero richiede 2 bytes di codice macchina. Il primo byte contiene il codice operativo, il modo di indirizzamento ed indica quale sarà il registro da usare come indice; il secondo byte contiene un indirizzo in pagina zero al quale verrà sommato il valore contenuto nel registro indice indicato al fine di ottenere l'indirizzo effettivo della locazione di memoria che contiene i dati con cui si intende operare.

Esempio: LDA $20,X
Se il registro indice X contiene il valore $3A, l'accumulatore viene caricato con il contenuto della locazione di memoria 005A o locazione $5A in pagina zero. Dopo l'assemblaggio avremo: B5 20
Notare che se il risultato dell'addizione supera il numero $FF non viene generato il riporto. Se nell'esempio il registro indice avesse contenuto il numero $F4, l'accumulatore sarebbe stato caricato con il contenuto della locazione $0014. Questo viene comunemente detto wrap-around (riavvolgimento).

Indirizzamento indiretto

L'indirizzamento indiretto è applicabile solo all'istruzione di salto (JMP). Esso richiede 3 bytes di codice macchina.
Il primo byte contiene il codice operativo, il secondo ed il terzo contengono l'indirizzo di un puntatore nella forma low-high (prima il byte meno significativo e poi il più significativo). L'indirizzo effettivo è contenuto nella cella di memoria indicata dal puntatore e dalla successiva, sempre nella forma low-high.

Esempio: JMP($2036)
Se il contenuto della locazione $2036 e $2037 sono $A9 e $54 rispettivamente il contatore di programma, dopo l'esecuzione dell'istruzione, conterrà $54A9.

2036	A9
2037	54

Indirizzamento indicizzato indiretto

L'indirizzamento indicizzato indiretto richiede 2 bytes di codice macchina. Il primo byte contiene il codice operativo; il secondo byte è sommato al contenuto del registro indice per formare un puntatore in pagina zero. Il byte meno significativo ed il byte più significativo dell'indirizzo effettivo sono contenuti nella locazio-

ne di memoria indicata dal puntatore e nella successiva. I dati su cui operare si trovano all'indirizzo effettivo così indicato.

Esempio: LDA ($54,X)
Se il registro X contiene il valore $38 ed il contenuto delle locazioni di memoria $009C e $009D sono rispettivamente $35 e $7B, l'accumulatore sarà caricato con il contenuto della locazione di memoria $7B35. Nella determinazione del puntatore non viene generato riporto (wrap-around). Solo il registro X può essere usato.

009C	35
009D	7B

Indirizzamento indiretto indicizzato

L'indirizzamento indiretto indicizzato richiede 2 bytes di codice macchina. Il primo byte contiene il codice operativo; il secondo byte contiene un indirizzo di pagina zero. Il contenuto di questa locazione di memoria è sommato al contenuto del registro Y per formare il byte meno significativo dell'indirizzo effettivo; l'eventuale riporto di questa somma è aggiunto al contenuto della successiva cella di memoria in pagina zero e forma il byte più significativo dell'indirizzo effettivo.
I dati sono contenuti all'indirizzo così calcolato.

Esempio: LDA ($A8),Y
Il contenuto della cella di memoria A8 e della successiva formano un indirizzo al quale verrà sommato il registro Y, che per esempio contiene $30, per ottenere l'indirizzo effettivo. Se le locazioni di memoria $A8 e $A9) contengono rispettivamente $37 e $15 il contenuto della locazione $1567 ($1537+$30) verrà caricato nell'accumulatore.

00A8	37
00A9	15

Indirizzamento implicito

L'indirizzamento implicito richiede un byte di codice macchina.

Esempio: CLC
L'indicatore di riporto viene azzerato. L'assemblaggio renderà 18.

Indirizzamento dell'accumulatore.

Questo indirizzamento richiede un byte di codice macchina; si applica alle operazioni che riguardano il contenuto dell'accumulatore. Esempio: ROL A
Il contenuto dell'accumulatore e dell'indicatore (flag) di riporto sono ruotati di un bit verso sinistra. Verrà assemblato con il codice 2A.

Indirizzamento relativo

L'indirizzamento relativo è usato esclusivamente con le istruzioni di salto condizionato e richiede 2 byte di codice macchina.
Quando viene eseguita una istruzione di salto condizionato lo stato di uno dei bit del registro di stato F viene controllato e la sua condizione (1 o 0) condiziona l'esecuzione del salto. Il primo byte dell'istruzione contiene il codice operativo ed il secondo un numero in complemento a due che rappresenta la distanza dalla posizione attuale del contatore di programma.

Esempio: BEQ $06
In linguaggio assembly si scrive:

Locazione	Codice macchina	Linguaggio assembly
PC	FO 06	BEQ $06
PC+2		

Se l'indicatore Z è impostato a 1 il contatore di programma è impostato a (PC+2)+6 = PC+8; se Z = 0 il contatore di programma resta a PC+2. Dato che c'è solo un byte a definire lo spostamento, la distanza di questo deve essere compresa tra -128_{10} e $+127_{10}$ dalla posizione corrente del contatore di programma. Se si vuole ottenere un salto condizionato al di fuori di questi valori si deve effetuare il salto ad una istruzione di JMP (salto incondizionato).

Tempi di esecuzione

Il tempo di esecuzione di ogni istruzione è indicato con il numero di cicli del clock (orologio) di sistema. La tavola I indica per ogni istruzione il numero di cicli richiesti, il numero varia al variare del modo di indirizzamento. Per le istruzioni di salto condizionato è indicato il tempo nel caso di salto non avvenga; se il salto viene eseguito è necessario aggiungere un ciclo se l'indirizzo è nella stessa pagina dell'istruzione, due cicli se si attraversa la pagina.

Esempio:

Locazione	Etichetta	Linguaggio assembly
2090		BPL LOOP
20A0	LOOP	

L'istruzione BPL controlla lo stato dell'indicatore N e provoca un salto se tale indicatore è a 0.
Se N=1 il tempo di esecuzione è di 2 cicli
Se N=0 il tempo di esecuzione è di 3 cicli (salto nella stessa pagina).
Se l'indirizzo della label (etichetta) fosse $2101 (un'altra pagina) e N=0 il tempo di esecuzione sarebbe di 4 cicli.
L'esame della tabella I indica che il tempo di esecuzione di alcune istruzioni per alcuni modi di indirizzamento varia se si attraversa la pagina di memoria.

Esempio: ciclo singolo di ritardo in una stessa pagina

Locazione	Etichetta	Linguaggio assembly	Commento
3000		LDX =$hh	; 2 cicli
3002	LOOP	DEX	; 2 cicli
3003		BNE LOOP	; 3 cicli per salto 2 altrimenti

Se hh è uguale a X_{10} il numero totale di cili e:
$2+2x + 3(x-1) + 2 = 5x + 1$
L'istruzione LDX è eseguita una sola volta, l'istruzione DEX x volte ed il salto si verificherà x-1 volte; 2 cicli devono essere inclusi per la condizione di non salto per BNE.
Altro esempio: doppio ciclo di ritardo con attraversamento di pagina per il ciclo esterno.

Locazione	Codice macchina	Etichetta	Assembly	Commento
2FFC	AØ hh		LDY #$hh	; 2 cicli
2FFE	A2 nn	LOOP2	LDX #$nn	; 2 cicli
3000	CA	LOOP1	DEX	; 2 cicli
3001	DØ FD		BNE LOOP1	; 3 o 2 cicli
3003	88		DEY	, 2 cicli
3004	DØ F8		BNE LOOP2	; 4 o 2 cicli

Il numero totale di cicli se hh=y_{10} e nn=x_{10} sarà: y(5x+7). Il periodo di ritardo è dato dal numero di cicli per il periodo del clock di sistema.

Paginazione

Il microprocessore 6502 può accedere fino a 65536 o 64K locazioni di memoria usando il suo bus di indirizzamento a 16 bit; gli indirizzi delle locazioni sono normalmente espressi con numeri esadecimali. L'intera gamma di indirizzamento è perciò compresa tra $0000 e $FFFF dove $ indica che si tratta di numeri esadecimali. Notare che 1K è equivalente a 1024_{10}.

La memoria del 6502 è organizzata in pagine; il byte più significativo dell'indirizzo assoluto è il numero di pagina, mentre il byte meno significativo indica la locazione all'interno della pagina. Per esempio l'indirizzo della locazione di memoria $3022 può essere espresso come locazione $22 della pagina $30. Ci sono quindi 256 pagine con 256 byte di memoria per pagina. Ci sono pochissime restrizioni imposte dal costruttore del microprocessore per il suo interfacciamento con la memoria ed i dispositivi di immissione/emissione (input/output).
Bisogna comunque notare che le pagine $00, $01 e $FF sono usate per scopi particolari.

Pagina $00: i vari modi di indirizzamento indicizzati usano questa pagina per memorizzare dati e indirizzi. Poiché il 6502 non ha registri indice a 16 bit la potenza del suo set di istruzioni sarebbe diminuita usando queste locazioni per il programma principale.

Pagina $01: L'area dello stack è limitata a questa pagina. Se il programma utente utilizza routine e/o interrupt, non bisognerebbe inserire istruzioni in questa pagina perché potrebbero essere danneggiate nel caso il puntatore dello stack le raggiungesse.

Pagina $FF: le ultime 6 locazioni di questa pagina sono riservate per gli indirizzi delle routine di interrupt e di reset. In molti sistemi a microcomputer questa pagina contiene il programma di monitor.

Legenda dei simboli delle tavole I e II

Accumulatore	A
Registro indice X	X
Registro indice Y	Y
Memoria	M
Registro di stato	P
Puntatore dello stack	S
Or esclusivo	⊕
Or logico	or

And logico	.
Somma	+
Sottrazione	-
Trasferimento verso	→
Trasferimento da	←
Push (inserimento) nello stack	↓
Pull (estrazione) dallo stack	↑
Contatore di programma	PC
Byte più significativo del PC	PC_h
Byte meno significativo del PC	PC_l
Operando	Operando
Immediato	#

Registro di stato

N	V		B	D	I	Z	C

NB: se la cornice di un indicatore è in grassetto significa che l'istruzione può variare lo stato dell'indicatore.

TAVOLA I ELENCO DELLE ISTRUZIONI

ADC: somma con riporto dell'accumulatore con la memoria

Operazione: A+M+C→A,C
Descrizione: il contenuto dell'indicatore di riporto è sommato al contenuto dell'accumulatore e della memoria; il risultato è posto nell'accumulatore.

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo del risultato
V: impostato a 1 se l'addizione ha reso un risultato, in complemento 2, maggiore di 127; altrimenti azzerato.
Z: impostato a 1 se il risultato è 0, altrimenti viene azzerato
C: Impostato a 1 se il risultato supera 255, a zero altrimenti

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Immediato	ADC #Operando	2	2	69
Pagina zero	ADC Operando	3	2	65
Pagina zero, X	ADC Operando,X	4	2	75
Assoluto	ADC Operando	4	3	6D
Assoluto,X	ADC Operando,X	4*	3	7D
Assoluto,Y	ADC Operando,Y	4*	3	79
(Indiretto,X)	ADC (Operando,X)	6	2	61
(Indiretto),Y	ADC (Operando),Y	5*	2	71

* Sommare 1 se si attraversa la pagina

AND: And della memoria con l'accumulatore

Operazione: A.M→A

Descrizione: Viene effettuato un and logico tra i bits della memoria ed i corrispondenti bits dell'accumulatore; il risultato è posto nell'accumulatore ed il contenuto della memoria resta invariato.

Registro di stato:

N	V		B	D	I	Z	C

N: Impostato come il bit più significativo del risultato

Z: Impostato a 1 se il risultato è zero, altrimenti azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Immediato	AND # Operando	2	2	29
Pagina zero	AND Operando	3	2	25
Pagina zero,X	AND Operando,X	4	2	35
Assoluto	AND Operando	4	3	2D
Assoluto,X	AND Operando,X	4*	3	3D
Assoluto,Y	AND Operando,Y	4*	3	39
(Indiretto,X)	AND (Operando,X)	6	2	21
(Indiretto),Y	AND (Operando),Y	5*	2	31

* Sommare 1 se si attraversa la pagina

ASL: Scorrimento (shift) a sinistra di 1 bit

Operazione: C← 7 6 5 4 3 2 1 0←0
Descrizione: il contenuto della memoria o dell'accumulatore è spostato verso sinistra di un bit; il bit 0 è caricato con uno zero logico ed il bit 7 è caricato nell'indicatore di riporto.

Registro di stato:

N	V		B	D	I	Z	C

N: impostato a uno se il bit più significativo della memoria o dell'accumulatore è a uno, azzerato altrimenti.
Z: impostato a uno se il risultato della memoria o dell'accumulatore è zero, azzerato altrimenti
C: prende il valore del bit 7 della memoria o dell'accumulatore

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Accumulatore	ASL A	2	1	0A
Pagina zero	ASL Operando	5	2	06
Pagina zero,X	ASL Operando,X	6	2	16
Assoluto	ASL Operando	6	3	0E
Assoluto,X	ASL Operando,X	7	3	1E

BCC: salta se l'indicatore di riporto è a zero

Descrizione: viene controllato l'indicatore C; se è a zero viene effettuato il salto, altrimenti si prosegue con la successiva istruzione.

Registro di stato
(non viene influenzato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Relativo	BCC Operando	2*	2	90

* Aggiungere 1 se si effettua il salto; aggiungere 2 se il salto provoca il cambio della pagina.

BCS: salta se l'indicatore di riporto è a uno

Descrizione: viene controllato l'indicatore C; se è a uno viene effettuato il salto, altrimenti si prosegue con la successiva istruzione.
Registro di stato:
(non viene influenzato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Relativo	BCS Operando	2*	2	B0

* Aggiungere 1 se si effettua il salto; aggiungere 2 se il salto provoca il cambio della pagina.

BEQ: salta se il risultato è zero

Descrizione: viene controllato l'indicatore di zero, se è impostato a 1 viene eseguito il salto altrimenti si procede in sequenza con la prossima istruzione.

Registro di stato:
(non viene influenzato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Relativo	BEQ Operando	2*	2	F0

*Aggiungere 1 se si effettua il salto; aggiungere 2 se il salto provoca il cambio della pagina.

BIT: controllo di bit

Operazione: A.M, $M_7 \rightarrow N$ e $M_6 \rightarrow V$
Descrizione: viene effettuato un AND tra i bit dell'accumulatore ed i rispettivi bit della memoria; il contenuto dell'accumulatore e della memoria resta invariato. I bit 6 e 7 della memoria sono trasferiti negli indicatori N e V rispettivamente.
Registro di stato:

N	V		B	D	I	Z	C

N: prende il valore del bit più significativo della memoria
Z: impostato a 1 se il risultato dell'operazione è zero
V: prende il valore del bit 6 della memoria

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Pagina zero	BIT Operando	3	2	24
Assoluto	BIT Operando	4	3	2C

BMI: salta se il risultato è negativo

Descrizione: viene controllato l'indicatore N, se è impostato a 1 viene eseguito il salto, altrimenti si procede in sequenza.

Registro di stato:
(non viene influenzato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Relativo	BMI Operando	2*	2	30

*Aggiungere 1 se si effettua il salto; aggiungere 2 se il salto provoca il cambio della pagina

BNE: salta se il risultato è diverso da zero

Descrizione: viene controllato l'indicatore Z, se è impostato a zero il salto viene eseguito; altrimenti si procede in sequenza.

Registro di stato:
(non viene influenzato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Relativo	BNE Operando	2*	2	D0

*Aggiungere 1 se si effettua il salto; aggiungere 2 se il salto provoca il cambio della pagina.

BPL: salta se il risultato è positivo

Descrizione: viene controllato l'indicatore N, se è impostato a 0 viene eseguito il salto, altrimenti si procede in sequenza.

Registro di stato:
(non viene influenzato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Relativo	BPL Operando	2*	2	10

*Aggiungere 1 se si effettua il salto; aggiungere 2 se il salto provoca il cambio della pagina

BRK: richiesta di interruzione (interrupt) da programma

Operazione: PC + 2↓, P↓
Descrizione: viene impostato a 1 l'indicatore di BREAK. Il contatore di programma viene incrementato di 2; il byte più significativo ed il meno significativo del contatore di programma vengono memorizzati nello stack, il registro di stato viene memorizzato nello stack. Il controllo del programma è trasferito alla routine di servizio dell'interrupt.

Registro di stato:

N	V		B	D	I	Z	C

B: impostato a 1
I: Impostato a 1

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	BRK	7	1	00

BVC: salta se l'indicazione di overflow (V) è a zero

Descrizione: viene controllato l'indicatore V, se è impostato a zero viene eseguito il salto, altrimenti si procede in sequenza.

Registro di stato:
(non viene influenzato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Relativo	BVC Operando	2*	2	50

*Aggiungere 1 se si effettua il salto; aggiungere 2 se il salto provoca il cambio della pagina

BVS: salta se l'indicatore di overflow (V) è impostato a 1

Descrizione: Viene controllato l'indicatore V, se è impostato a 1 viene eseguito il salto, altrimenti si procede in sequenza.

Registro di stato:
(non viene influenzato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Relativo	BVS Operando	2*	2	70

* Aggiungere 1 se si effettua il salto; aggiungere 2 se il salto provoca il cambio della pagina

CLC: azzera l'indicatore di riporto

Operazione: 0 → C

Descrizione: l'indicatore di riporto viene azzerato

Registro di stato:

N	V		B	D	I	Z	C

C: azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	CLC	2	1	18

CLD: azzera l'indicatore di decimale

Operazione: 0 → D

Descrizione: viene azzerato l'indicatore di decimale. Tutte le operazioni aritmetiche verranno eseguite in binario.

Registro di stato:

N	V		B	D	I	Z	C

D: azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	CLD	2	1	D8

CLI: attiva le richieste di IRQ

Operazione: 0 → I

Descrizione: Abilita la ricezione delle richieste di IRQ

Registro di stato:

N	V		B	D	I	Z	C

I: azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	CLI	2	1	58

CLV: azzera l'indicatore di overflow (V)

Operazione: Ø → V

Descrizione: l'indicatore di overflow è azzerato

Registro di stato:

V: azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	CLV	2	1	B8

CMP: confronto tra memoria e accumulatore

Operazione: A - M

Descrizione: confronto, mediante sottrazione, del contenuto della memoria con quello dell'accumulatore; il contenuto dell'accumulatore e della memoria resta invariato.

Registro di stato

N	V		B	D	I	Z	C

N: impostato come il bit più significativo del risultato della comparazione

Z: impostato a 1 se il risultato del confronto è zero, azzerato se il risultato è diverso da zero

C: azzerato se il contenuto della memoria è maggiore di quello dell'accumulatore; altrimenti impostato a 1

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Immediato	CMP #Operando	2	2	C9
Pagina zero	CMP Operando	3	2	C5
Pagina zero, X	CMP Operando, X	4	2	D5
Assoluto	CMP Operando	4	3	CD
Assoluto, X	CMP Operando, X	4*	3	DD
Assoluto, Y	CMP Operando, Y	4*	3	D9
(Indiretto, X)	CMP (Operando, X)	6	2	C1
(Indiretto), Y	CMP (Operando), Y	5*	2	D1

*Aggiungere 1 se si attraversa il limite di pagina

CPX: confronto tra memoria ed il registro indice X

Operazione: X - M

Descrizione: confronto, mediante sottrazione, del contenuto della memoria con quello del registro indice X; il contenuto della memoria e del registro indice X resta invariato.

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo del risultato della comparazione

Z: impostato a 1 se il risultato del confronto è zero, azzerato se il risultato è diverso da zero

C: azzerato se il contenuto della memoria è maggiore del contenuto del registro indice X; altrimenti impostato a 1

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Immediato	CPX #Operando	2	2	E0
Pagina zero	CPX Operando	3	2	E4
Assoluto	CPX Operando	4	3	EC

CPY: confronto tra memoria e registro Y

Operazione: Y - M

Descrizione: confronto, mediante sottrazione, del contenuto della memoria con quello del registro indice Y; il contenuto della memoria e del registro indice Y resta invariato.

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo del risultato della comparazione
Z: impostato a 1 se il risultato del confronto è zero, azzerato e il risultato è diverso da zero
C: azzerato se il contenuto della memoria è maggiore del contenuto del registro indice Y; altrimenti impostato a 1

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Immediato	CPX #Operando	2	2	C0
Pagina zero	CPX Operando	3	2	C4
Assoluto	CPX Operando	4	3	CC

DEC: decremento della memoria

Operazione: M-1 → M

Descrizione: il contenuto della memoria è decrementato di 1

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo della memoria
Z: impostato a 1 se il contenuto della memoria è zero

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Pagina zero	DEC Operando	5	2	C6
Pagina zero, X	DEC Operando, X	6	2	D6
Assoluto	DEC Operando	6	3	CE
Assoluto, X	DEC Operando, X	7	3	DE

DEX: decremento del registro indice X

Operazione: X-1 → X

Descrizione: il contenuto del registro indice è decrementato di 1

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo del registro
Z: impostato a 1 se il contenuto del registro è zero: altrimenti viene azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	DEX	2	1	CA

DEY: decremento del registro indice Y

Operazione: Y-1 → Y
Descrizione: il contenuto del registro indice è decrementato di 1

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo del registro
Z: impostato a 1 se il contenuto del registro è zero: altrimenti viene azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	DEY	2	1	88

EOR: OR esclusivo della memoria con l'accumulatore

Operazione: $A \oplus M \rightarrow A$

Descrizione: viene effettuato un OR esclusivo tra i bit della memoria ed i rispettivi bit dell'accumulatore; il risultato viene posto nell'accumulatore; il contenuto della memoria resta invariato.

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo del risultato
Z: impostato a 1 se il risultato è uguale a zero; altrimenti viene azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Immediato	EOR #Operando	2	2	49
Pagina zero	EOR Operando	3	2	45
Pagina zero, X	EOR Operando, X	4	2	55
Assoluto	EOR Operando	4	3	4D
Assoluto, X	EOR Operando, X	4*	3	5D
Assoluto, Y	EOR Operando, Y	4*	3	59
(Indiretto, X)	EOR (Operando, X)	6	2	41
(Indiretto), Y	EOR (Operando), Y	5*	2	51

INC: incremento della memoria

Operazione: M + 1 → M

Descrizione: il contenuto della memoria viene incrementato di 1

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo del risultato
Z: impostato a 1 se il risultato è uguale a zero; altrimenti viene azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Pagina zero	INC Operando	5	2	E6
Pagina zero, X	INC Operando, X	6	2	F6
Assoluto	INC Operando	6	3	EE
Assoluto, X	INC Operando, X	7	3	FE

INX: incremento del registro indice X

Operazione: X + 1 → X

Descrizione: il contenuto del registro viene incrementato di 1

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo del risultato
Z: impostato a 1 se il risultato è uguale a zero; altrimenti viene azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	INX	2	1	E8

INY: incremento del registro indice Y

Operazione: $Y + 1 \rightarrow Y$

Descrizione: il contenuto del registro viene incrementato di 1

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo del risultato
Z: impostato a 1 e il risultato è uguale a zero; altrimenti viene azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	INY	2	1	C8

JMP: salto incondizionato

Operazione: $(PC + 1) \rightarrow PC_{low}$, $(PC + 2) \rightarrow PC_h$

Descrizione: salto all'indirizzo indicato dall'operando

Registro di stato:
(non viene modificato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Assoluto	JMP Operando	3	3	4C
Indiretto	JMP (Operando)	5	3	6C

JSR: salto a subroutine

Operazione: $PC + 2 \rightarrow$, $(PC + 1) \rightarrow PC_l$ $(PC + 2) \rightarrow PC_h$

Descrizione: il contatore di programma viene incrementato di 2. Il byte più significativo del PC viene copiato nello stack, lo SP viene decrementato, il byte meno significativo del PC viene copiato nello stack e lo SP viene nuovamente decrementato. Si effettua un salto all'indirizzo contenuto nell'operando.

Registro di stato:
(non viene modificato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Assoluto	JSR Operando	6	3	20

LDA: carica l'accumulatore

Operazione: M → A
Descrizione: il contenuto della memoria è copiato nell'accumulatore

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo del risultato
Z: impostato a 1 se il risultato è uguale a zero; altrimenti viene azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Immediato	LDA #Operando	2	2	A9
Pagina zero	LDA Operando	3	2	A5
Pagina zero, X	LDA Operando, X	4	2	B5
Assoluto	LDA Operando	4	3	AD
Assoluto, X	LDA Operando, X	4*	3	BD
Assoluto, Y	LDA Operando, Y	4*	3	B9
(Indiretto, X)	LDA (Operando, X)	6	2	A1
(Indiretto), Y	LDA (Operando), Y	5*	2	B1

*Aggiungere 1 se si attraversa il limite di pagina

LDX: carica il registro indice X

Operazione: M → X

Descrizione: il contenuto della memoria è copiato nel registro

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo del risultato
Z: impostato a 1 se il risultato è uguale a zero; altrimenti viene azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Immediato	LDX #Operando	2	2	A2
Pagina zero	LDX Operando	3	2	A6
Pagina zero, Y	LDX Operando, Y	4	2	B6
Assoluto	LDX Operando	4	3	AE
Assoluto, Y	LDX Operando, Y	4*	3	BE

*Aggiungere 1 se si attraversa il limite di pagina

LDY: carica il registro indice Y

Operazione: M → Y

Descrizione: il contenuto della memoria è copiato nel registro

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo del risultato
Z: impostato a 1 se il risultato è uguale a zero; altrimenti viene azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Immediato	LDY #Operando	2	2	AØ
Pagina zero	LDY Operando	3	2	A4
Pagina zero, X	LDY Operando, X	4	2	B4
Assoluto	LDY Operando	4	3	AC
Assoluto, X	LDY Operando, X	4*	3	BC

*Aggiungere 1 se si attraversa il limite di pagina

LSR: scorrimento a destra di 1 bit
Operazione: 0 → 7 6 5 4 3 2 1 0 → C

Descrizione: il contenuto dell'accumulatore o della memoria è spostato verso destra di un bit; il bit più significativo viene impostato a zero ed il bit meno significativo viene mosso nell'indicatore di riporto.

Registro di stato:

N	V		B	D	I	Z	C

N: azzerato
Z: impostato a 1 se il contenuto della memoria o dell'accumulatore è uguale a zero; altrimenti viene azzerato
C: prende il valore del bit meno significativo

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Accumulatore	LSR A	2	1	4A
Pagina zero	LSR Operando	5	2	46
Pagina zero, X	LSR Operando, X	6	2	56
Assoluto	LSR Operando	6	2	4E
Assoluto, X	LSR Operando, X	7	3	5E

NOP: nessuna operazione

Descrizione: per due cicli non viene eseguita nessuna operazione

Registro di stato:
(non viene modificato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	NOP	2	1	EA

ORA: OR tra memoria e accumulatore

Operazione: A or M → A

Descrizione: viene effettuata l'operazione di OR tra i bits della memoria ed i rispettivi bit dell'accumulatore; il risultato viene posto nell'accumulatore. Il contenuto della memoria resta invariato.

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo del risultato
Z: impostato a 1 se il risultato è uguale a zero; altrimenti viene azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Immediato	ORA #Operando	2	2	09
Pagina zero	ORA Operando	3	2	05
Pagina zero, X	ORA Operando, X	4	2	15
Assoluto	ORA Operando	4	3	0D
Assoluto, X	ORA Operando, X	4*	3	1D
Assoluto, Y	ORA Operando, Y	4*	3	19
(Indiretto, X)	ORA (Operando, X)	6	2	01
(Indiretto), Y	ORA (Operando), Y	5*	2	11

*Aggiungere 1 se si attraversa il limite di pagina

PHA: copia l'accumulatore nello stack

Operazione: A ↓

Descrizione: il contenuto dell'accumulatore è copiato nello stack all'indirizzo puntato da SP, SP viene poi decrementato di 1

Registro di stato:
(non viene modificato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	PHA	3	1	48

PHP: copia il registro di stato nello stack

Operazione: P ↓

Descrizione: il registro di stato P viene copiato nello stack all'indirizzamento puntato da SP; SP viene poi decrementato di 1

Registro di stato:
(non viene modificato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	PHP	3	1	08

PLA: carica l'accumulatore dallo stack

Operazione: A ↑

Descrizione: SP viene incrementato di 1; l'accumulatore è caricato con la locazione di memoria puntata da SP.

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo dell'accumulatore
Z: impostato a 1 e il contenuto dell'accumulatore è zero; altrimenti viene azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	PLA	4	1	68

PLP: carica il registro di stato dallo stack

Operazione: P ↑

Descrizione: SP viene incrementato di 1; il registro di stato è caricato con il contenuto della locazione di memoria puntata da SP (stack pointer)

Registro di stato:

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	PLP	4	1	28

ROL: rotazidone a sinistra

Operazione: C → 7 6 5 4 3 2 1 0 → C

Descrizione: Il contenuto della memoria o dell'accumulatore è ruotato a sinistra di 1 bit; l'indicatore di riporto è ruotato nel bit 0 ed il bit più significativo è ruotato nell'indicatore di riporto.

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo della memoria o dell'accumulatore dopo la rotazione
Z: impostato a 1 se, dopo la rotazione, il contenuto della memoria o dell'accumulatore è zero; altrimenti azzerato
C: impostato come il bit più significativo della memoria o dell'accumulatore, prima della rotazione

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Accumulatore	ROL A	2	1	2A
Pagina zero	ROL Operando	5	2	26
Pagina zero, X	ROL Operando, X	6	2	36
Assoluto	ROL Operando	6	3	2E
Assoluto, X	ROL Operando, X	7	3	3E

ROR: rotazione a destra

Operazione: C ← 7 6 5 4 3 2 1 0 ← C

Descrizione: Il contenuto della memoria o dell'accumulatore è ruotato verso destra di 1 bit; l'indicatore di riporto è ruotato nel bit più significativo della memoria o dell'accumulatore, il bit 0 è ruotato nell'indicatore di riporto.

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo dpo la rotazione
Z: impostato a 1 se, dopo la rotazione, il contenuto della

memoria o dell'accumulatore è zero; altrimenti azzerato
C: impostato come il bit meno significativo prima della rotazione

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Accumulatore	ROR A	2	1	6A
Pagina zero	ROR Operando	5	2	66
Pagina zero, X	ROR Operando, X	6	2	76
Assoluto	ROR Operando	6	3	6E
Assoluto, X	ROR Operando, X	7	3	7E

RTI: ritorno da interruzione

Operazione: P↑ PC↑

Descrizione: il puntatore dello stack (SP) è incrementato di 1. Il registro di stato è caricato con il byte puntato da SP; poi SP viene incrementato di uno ed il byte così puntato viene caricato nel byte meno significativo del contatore di programma, SP viene ancora incrementato ed il byte puntato viene caricato nel byte più significativo del PC.

Registro di stato:

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	RTI	6	1	40

RTS: ritorno da subroutine

Operazione: PC↑, PC+1 → PC

Descrizione: il puntatore dello stack (SP) è incrementato di 1. Il byte puntato da SP è caricato nel byte meno significativo del contatore di programma (PC); SP viene ancora incrementato di 1 ed byte così puntato viene caricato nel byte più significativo del PC. Il PC è quindi incrementato di 1.

Registro di stato:
(non viene modificato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	RTS	6	1	60

SBC: sottrazione con riporto

Operazione: $A - M - \overline{C} \rightarrow A,C$

Descrizione: il contenuto della memoria più il complemento dell'indicatore di riporto è sottratto dal contenuto dell'accumulatore; il risultato è posto nell'accumulatore.

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo del risultato
Z: impostato a 1 e il risultato è zero; altrimenti azzerato
C: azzerato se il contenuto della memoria più il complemento dell'indicazione di riporto è maggiore del contenuto dell'accumulatore; altrimenti impostato a 1
V: impostato a 1 se il risultato genera un'eccedenza considerando il numero in complemento a due.

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Immediato	SBC ≠Operando	2	2	E9
Pagina zero	SBC Operando	3	2	E5
Pagina zero,X	SBC Operando,X	4	2	F5
Assoluto	SBC Operando	4	3	ED
Assoluto,X	SBC Operando,X	4*	3	FD
Assoluto,Y	SBC Operando,Y	4*	3	F9
(Indiretto,X)	SBC (Operando,X)	6	2	E1
(Indiretto),Y	SBC (Operando),Y	5*	2	F1

*Aggiungere 1 se si attraversa la pagina

SEC: imposta a uno l'inndicatore di riport

Operazione: 1 → C

Descrizione: l'indicatore di riporto è impostato a uno

Registro di stato

N	V		B	D	I	Z	C

C: impostato a 1

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	SEC	2	1	38

SED: imposta il funzionamento decimale

Operazione: 1 → D

Descrizione: l'indicatore di decimale è impostato a 1

Registro di stato:

N	V		B	D	I	Z	C

D: impostato a 1

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	SED	2	1	F8

SEI: disattivo le richieste di IRQ

Operazione: 1 → I

Descrizione: l'indicatore di interrupt disabilitato viene impostato a 1

Registro di stato:

N	V		B	D	I	Z	C

I: impostato a 1

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	SEI	2	1	78

STA: copia l'accumulatore in memoria

Operazione: A → M

Descrizione: il contenuto dell'accumulatore è copiato in una locazione di memoria

Registro di stato:
(non viene modificato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Pagina zero	STA Operando	3	2	85
Pagina zero,X	STA Operando,X	4	2	95
Assoluto	STA Operando	4	3	8D
Assoluto,X	STA Operando,X	5	3	9D
Assoluto,Y	STA Operando,Y	5	3	99
(Indiretto,X)	STA (Operando,X)	6	2	81
(Indiretto),Y	STA (Operando),Y	6	2	91

STX: copia di registro indice X in memoria

Operazione: X → M

Descrizione: il contenuto del registro indice X è copiato in una locazione di memoria.

Registro di stato:
(non viene modificato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Pagina zero	STA Operando	3	2	86
Pagina zero,Y	STX Operando,Y	4	2	96
Assoluto	STX Operando	4	3	8E

STY: copia il registro indice Y in memoria

Operazione: Y → M

Descrizione: il contenuto del registro indice Y è copiato in una locazione di memoria

Registro di stato:
(non viene modificato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Pagina zero	STY Operando	3	2	84
Pagina zero,X	STY Operando,X	4	2	94
Assoluto	STY Operando	4	3	8C

TAX: copia il contenuto dell'accumulatore nel registro X

Operazione: A → X

Descrizione: il contenuto dell'accumulatore è copiato nel registro indice X. Il contenuto dell'accumulatore resta invariato.

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo dell'accumulatore
Z: impostato a uno se il contenuto dell'accumulatore è zero, altrimenti azzerato.

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	TAX	2	2	AA

TAY: copia il contenuto dell'accumulatore nel registro Y

Operazione: A → Y

Descrizione: il contenuto dell'accumulatore è copiato nel registro indice Y. Il contenuto dell'accumulatore resta invariato.

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo dell'accumulatore
Z: impostato a uno se il contenuto dell'accumulatore è zero, altrimenti azzerato.

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	TAY	2	1	A8

TYA: copia il contenuto del registro Y nell'accumulatore

Operazione: Y → A

Descrizione: il contenuto del registro indice Y è copiato nell'accumulatore. Il contenuto del registro Y resta invariato.

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo dell'accumulatore
Z: impostato a uno e il contenuto dell'accumulatore è zero.

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	TYA	2	1	98

TSX: copia il contenuto di SP in X

Operazione: S → X

Descrizione: il contenuto dello stack pointer è copiato nel registro X, il contenuto di SP resta invariato

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo di SP
Z: impostato a uno se SP è uguale da zero, altrimenti azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	TSX	2	1	BA

TXA: copia il contenuto del registro X nell'accumulatore

Operazione: X → A

Descrizione: il contenuto del registro indice X è copiato nell'accumulatore, il contenuto del registro X resta invariato.

Registro di stato:

N	V		B	D	I	Z	C

N: impostato come il bit più significativo del registro X
Z: impostato a uno e il contenuto del registro X è zero, altrimenti azzerato

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	TXA	2	1	8A

TXS: copia il contenuto del registro X nel puntatore di stack

Operazione: X → S

Descrizione: il contenuto del registro indice X è copiato in SP, il contenuto del registro tra X resta invariato.

Registro di stato:
(non viene modificato)

N	V		B	D	I	Z	C

Modo di indirizzamento	Assembly	N. cicli	N. bytes	codice operativo
Implicito	TXS	2	1	9A

Tavola II Codici operativi in sequenza numerica

Cod.	Istruz.	Modo di Indirizzamento	Cod.	Istruz.	Modo di indirizzamento
00	BRK	Implied	28	PLP	Implied
01	ORA	(Indirect,X)	29	AND	Immediate
05	ORA	Zero page	2A	ROL	Accumulator
06	ASL	Zero page	2C	BIT	Absolute
08	PHP	Implied	2D	AND	Absolute
09	ORA	Immediate	2E	ROL	Absolute
0A	ASL	Accumulator	30	BMI	Relative
0D	ORA	Absolute	31	AND	(Indirect),Y
0E	ASL	Absolute	35	AND	Zero page,X
10	BPL	Relative	36	ROL	Zero page,X
11	ORA	(Indirect),Y	38	SEC	Implied
15	ORA	Zero page,X	39	AND	Absolute,Y
16	ASL	Zero page,X	3D	AND	Absolute,X

(segue)

(seguito)

18	CLC	Implied	3E	ROL	Absolute,X
19	ORA	Absolute,Y	40	RTI	Implied
1D	ORA	Absolute,X	41	EOR	(Indirect,X)
1E	ASL	Absolute,X	45	EOR	Zero page
20	ISR	Absolute	46	LSR	Zero page
21	AND	(Indirect,X)	48	PHA	Implied
24	BIT	Zero page	49	EOR	Immediate
25	AND	Zero page	4A	LSR	Accumulator
26	ROL	Zero page	4C	IMP	Absolute
4D	EOR	Absolute	99	STA	Absolute,Y
4E	LSR	Absolute	9A	TXS	Implied
50	BVC	Relative	9D	STA	Absolute,X
51	EOR	(Indirect),Y	A0	LDY	Immediate
55	EOR	Zero page,X	A1	LDA	(Indirect,X)
56	LSR	Zero page,X	A2	LDX	Immediate
58	CLI	Implied	A4	LDY	Zero page
59	EOR	Absolute,Y	A5	LDA	Zero page
5D	EOR	Absolute,X	A6	LDX	Zero page
5E	LSR	Absolute,X	A8	TAY	Implied
60	RTS	Implied	A9	LDA	Immediate
61	ADC	(Indirect,X)	AA	TAX	Implied
65	ADC	Zero page	AC	LDY	Absolute
66	ROR	Zero page	AD	LDA	Absolute
68	PLA	Implied	AE	LDX	Absolute
69	ADC	Immediate	B0	BCS	Relative
6A	ROR	Accumulator	B1	LDA	(Indirect,)Y
6C	IMP	Indirect	B4	LDY	Zero page,X
6D	ADC	Absolute	B5	LDA	Zero page,X
6E	ROR	Absolute	B6	LDX	Zero page,Y
70	BVS	Relative	B8	CLV	Implied
71	ADC	(Indirect),Y	B9	LDA	Absolute,Y
75	ADC	Zero page,X	BA	TSX	Implied
76	ROR	Zero page,X	BC	LDY	Absolute,X
78	SEI	Implied	BD	LDA	Absolute,X
79	ADC	Absolute,Y	BE	LDX	Absolute,Y

(segue)

(seguito)

7D	ADC	Absolute,X	C0	CPY	Immediate
7E	ROR	Absolute,X	C1	CMP	(Indirect,X)
81	STA	(Indirect,X)	C4	CPY	Zero page
84	STY	Zero page	C5	CMP	Zero page
85	STA	Zero page	C6	DEC	Zero page
86	STX	Zero page	C8	INY	Implied
88	DEY	Implied	C9	CMP	Immediate
8A	TXA	Implied	CA	DEX	Implied
8C	STY	Absolute	CC	CPY	Absolute
8D	STA	Absolute	CD	CMP	Absolute
8E	STX	Absolute	CE	DEC	Absolute
90	BCC	Relative	D0	BNE	Relative
91	STA	(Indirect),Y	D1	CMP	(Indirect),Y
94	STY	Zero page,X	D5	CMP	Zero page,X
95	STA	Zero page,X	D6	DEC	Zero page,X
96	STX	Zero page,Y	D8	CLD	Implied
98	TYA	Implied	D9	CMP	Absolute,Y
DD	CMP	Absolute,X	ED	SBC	Absolute
DE	DEC	Absolute,X	EE	INC	Absolute
E0	CPX	Immediate	F0	BEQ	Relative
E1	SBC	(Indirect,X)	F1	SBC	(Indirect),Y
E4	CPX	Zero page	F5	SBC	Zero page,X
E5	SBC	Zero page	E6	INC	Zero page,X
E6	INC	Zero page	F8	SED	Implied
E8	INX	Implied	F9	SBC	Absolute,Y
E9	SBC	Immediate	FD	SBC	Absolute,X
EA	NOP	Implied	FE	INC	Absolute,X
EC	CPX	Absolute			

FUNZIONAMENTO DELLO STACK

Lo stack è un'area di memoria dove la CPU e/o il programmatore può memorizzare temporaneamente dei dati. Nel sistema utiliz-

zante il 6502 lo stack è limitato alla pagina 01 della memoria e, poiché una pagina è costituita da 256 bytes, è sufficiente un registro a 8 bits per indicare la posizione corrente dello stack. Questo registro viene chiamato stack pointer (puntatore dello stack). Lo stack funziona nel seguente modo: i dati vengono inseriti un byte alla volta all'indirizzo indicato da SP (stack pointer); ogni volta che un byte viene inserito SP viene automaticamente decrementato di uno per puntare alla prossima locazione da utilizzare. Durante il prelievo dei dati viene prima incrementato SP, e poi prelevato il byte puntato da SP. La figura 1 descrive sinteticamente l'operazione. Supponiamo che inizialmente lo stack pointer contenga il valore SP. I dati vengono memorizzati in questa sequenza: BYTE1, BYTE2... BYTE5 mentre lo stack pointer viene decrementato fino a raggiungere il valore SP-5. Il prelievo dei dati avviene poi in sequenza inversa BYTE5... BYTE1 riportando lo stack al valore SP. Questo tipo di funzionamento viene comunemente detto LIFO (last-in-first-out).

SP-5	
SP-4	BYTE5
SP-3	BYTE4
SP-2	BYTE3
SP-1	BYTE2
SP	BYTE1

Caricamento dello stack pointer

Durante la sequenza di inizializzazione di un programma bisognerà provvedere ad impostare lo stack pointer con un indirizzo opportuno. Non ci sono istruzioni che caricano direttamente lo stack pointer

```
LDX ≠$FF      ; carica il registro X con $FF
TXS           ; trasferisce il contenuto di X nello stack
              pointer
```

La posizione corrente dello stack sarà così $01FF. Il registro Y non può essere usato per questa operazione.

Operazioni di «push» e di «pull»

Ci sono quattro istruzioni per inserire (push) e prelevare (pull) dati dallo stack. Esse sono: PHA, PHP, PLA e PLP

PHA: il contenuto dell'accumulatore è copiato nello stack e lo stack pointer è decrementato di 1.

PLA: lo stack pointer è incrementato di uno ed il dato così indirizzato è copiato nell'accumulatore.

PHP: il contenuto del registro di stato è copiato nello stack e lo stack pointer viene decrementato di uno.

PLP: lo stack pointer è incrementato di uno ed il dato così indirizzato è copiato nel registro di stato.

I contenuti dei registri X e Y non possono essere salvati direttamente nello stack ma devono essere precedentemente copiati nell'accumulatore. Esempio:

```
TXA     ; il contenuto del registro X è copiato in A
PHA     ; è memorizzato nello stack
```

Per riottenere il valore di X si eseguirà:

```
PLA     ; copiatura nell'accumulatore del dato
TAX     ; e copiatura in x
```

Usando le istruzioni TYA e TAY è quindi possibile salvare anche il registro indice Y.

Subroutine

Il controllo del programma viene trasferito ad una routine, (sottoprogramma) mediante l'istruzione JSR. Questa istruzione richiede tre bytes di codice macchina ed usa il modo di indirizzamento assoluto.

Esempio:

```
2200    JSR $0300
2203    -----
```

SP-2	
SP-1	02
SP	22

Quando l'istruzione viene eseguita il contatore di programma è incrementato di due ed il suo contenuto è inserito nello stack un byte alla volta dalla CPU. Notare che lo stack pointer è decrementato di due e che il byte più significativo è memorizzato per primo. Il controllo del programma è poi trasferito a $0300.
Le subroutine terminano normalmente con l'istruzione RTS.Questa è una istruzione di un byte e quando viene eseguita lo stack pointer è incrementato di 1, il byte puntato è copiato nel byte meno significativo del contatore di programma, lo stack pointer è ancora incrementato di uno ed il byte così puntato viene copiato nella parte più significativa del PC.
Infine il PC viene incrementato di uno ed il programma riprende dall'indirizzo ottenuto.
Nell'esempio precedente il programma riprenderebbe con la istruzione che si trova all'indirizzo 2203.
Fare attenzione che l'ingresso in una subroutine non salva automaticamente il registro di stato della CPU. Alcune volte bisognerà provvedere quindi al salvataggio di uno o più registri con le istruzioni viste prima.

Subroutine nidificate

È ammesso l'uso di subroutine nidificate

Esempio

```
2033          JSR SUB1   ;programma
2036
2080   SUB1               ;subroutine 1
2090          JSR SUB2
              RTS
3000   SUB2               ;subroutine 2
              RTS
```

Immediatamente prima dell'esecuzione di RTS nella subroutine 2 lo stack conteneva:

SP-4	
SP-3	92
SP-2	20
SP-1	35
SP	20

Interrupt

Quando viene ricevuta una richiesta di interrupt la CPU memorizza l'indirizzo di ritorno del programma principale (prima il byte più significativo) ed il contenuto del registro di stato (P) nello stack del seguente ordine:
Esempio: 2050 LDA ≠MEM ← ricezione dell'interrupt

SP-3	
SP-2	P
SP-1	52
SP	20

Alla ricezione di un interrupt viene completata l'istruzione corrente ed il contatore di programma conterrà l'indirizzo della prossima istruzione.
Le routine di servizio degli interrupt terminano, normalmente, con l'istruzione di RTI. Questa è una istruzione di 1 byte e quando viene eseguita provvede a. incrementare lo stack pointer, prelevare il contenuto del registro di stato e del contatore di programma dello stack. Nell'esempio precedente il programma riprenderebbe dalla locazione $2052.

Modifica della stack

I dati memorizzati nello stack possono essere modificati dal programmatore incrementando o decrementando lo stack pointer del valore richiesto ed usando istruzioni di push e pull.
Non ci sono istruzioni che incrementano o decrementano direttamente lo stack pointer ma questo può essere ottenuto usando il registro X.
TSX
DEX o INX
TXS
Questa routine copia il contenuto dello stack pointer nel registro X, il quale viene successivamente decrementato, o incrementato come appropriato, e ricopiato in SP.
Lo stack pointer può essere modificato anche con istruzioni di push e pull; bisogna prestare particolare attenzione a non danneggiare il contenuto dello SP o dell'accumulatore.

Immissione/Emissione

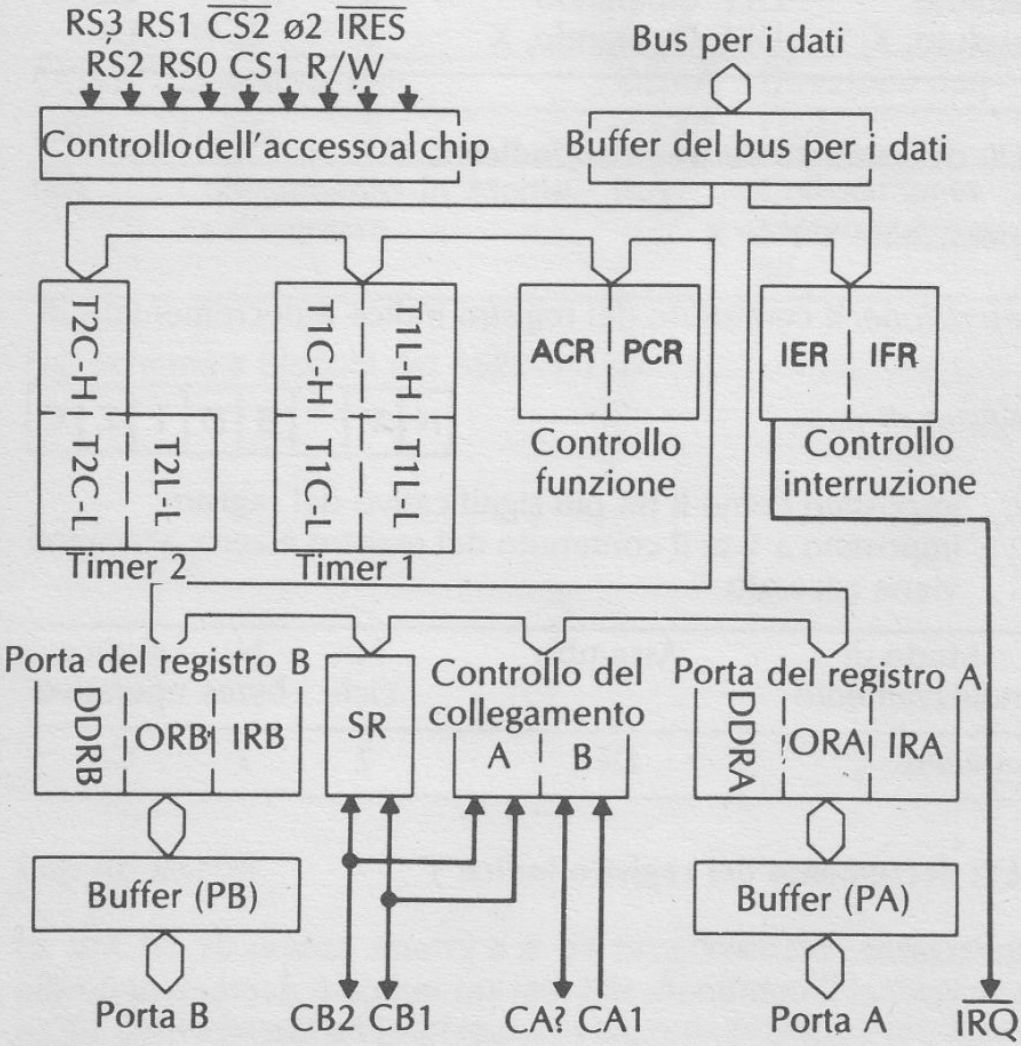
Ci sono molti dispositivi periferici compatibili con la CPU 6502. Esula dallo scopo della presente guida fornire la lista ed i dettagli di questi dispositivi. Di seguito vengono elencati solo i più diffusi con particolare riferimento alla VIA 6522.

(a) PERIPHERAL INTERFA CE ADAPTER (PIA) 6520
(b) ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA) 6551
(c) CRT CONTROLLER (CRTC) 6545
(d) ROM-RAM-I/O-COUNTER (RRIOC) 6531
(e) ROM-I/O-COUNTER (RIOC) 6534
(f) ROM-RAM-I/O-INTERVAL TIMER (RRIOT) 6530
(g) RAM, I/O,INTERVAL TIMER (RIOT) 6532
(h) VERSATILE INTERFACE ADAPTER (VIA) 6522

Per le esatte caratteristiche si rimanda ai fogli dati forniti dai vari costruttori.

Adattatore per interfaccia versatile (VIA) 6522

La VIA dispone delle seguenti caratteristiche:
a) due porte bidirezionali a 8 bit
b) due timer
c) registro per conversioni seriale/parallelo (shift register)
d) quattro linee di controllo CA1, CA2, CB1 e CB2
e) generazione di interrupt
La figura 2 indica lo schema a blocchi della VIA 6522, i suoi registri interni ed i terminali per il collegamento



Key			
T2C-H	Contatore	ACR	Ausiliario
T2C-L	Contatore	PCR	Periferico
T2L-L	Collegamento in Memoria	IER	Abilitare
T1C-L	Contatore	IFR	Flag
T1C-L	Contatore	SR	Registro shift
T1L-H	Collegamento in Memoria	A	Porta A
T1L-L	Collegamento in Memoria	B	Porta B
DDRB	Direzione dati	DDRA	Direzione dati
ORB	Uscita	ORA	Uscita
IRB	Collegamento in Memoria d'ingresso	IRA	Collegamento in Memoriad'ingresso

Fig. Schema a blocchi del 6522

Registri interni

La VIA ha 16 registri interni a 8 bit programmabili mappati in memoria. Il chip è selezionato mediante due linee CSI (attiva alta) e CS2 (attiva bassa) ed i vari registri vengono indirizzati mediante

4 linee RS0, RS1, RS2 e RS3. La tavola seguente mostra come indirizzare ogni registro.

Registro No. $	RS3	RS2	RS1	RS0	Registro Mnemonico	Descrizione: Scrittura	Descrizione: Lettura
0	0	0	0	0	ORB/IRB	Registro d'uscita B	Registro d'ingresso B
1	0	0	0	1	ORA/IRA	Registro d'uscita A	Registro d'ingresso A
2	0	0	1	0	DDRB	Registri direzione dati B	
3	0	0	1	1	DDRA	Registri direzione dati A	
4	0	1	0	0	T1C-L	Parte bassa del collegamento in MemoriaT1	Parte bassa del contatore T1
5	0	1	0	1	T1C-H	Parte alta del contatore T1	
6	0	1	1	0	T1L-L	Parte bassa del collegamento in Memoria T1	
7	0	1	1	1	T1L-H	Parte alta del collegamento in Memoria T1	
8	1	0	0	0	T2C-L	Parte bassa del collegamento in Memoria T2	Parte bassa del contatore T2
9	1	0	0	1	T2C-H	Parte alta del contatore T2	
A	1	0	1	0	SR	Registro shift	
B	1	0	1	1	ACR	Registro ausiliario di controllo	
C	1	1	0	0	PCR	Registro controllo periferiche	
D	1	1	0	1	IFR	Registro del flag per le interruzioni	
E	1	1	1	0	IER	Registro abilitazione interruttori	
F	1	1	1	1	ORA/IRA	Comunicazione senza protocollo	

NB: quando la VIA viene resettata, i registri DDRA e DDRB, i registri di controllo PCR e ACR e i registri degli interrupt IFR sono azzerati. Gli interrupt sono quindi disabilitati.

Porte di ingresso/uscita

Ogni Porta ha 8 linee (PA7-PAO, PB7-PBO) che possono essere configurate individualmente come ingressi o uscite mediante i registri di direzione DDRA e DDRB.

Registri di direzione DDRB e DDRA (registri $2 e $3)

Per definire una linea come ingresso bisogna scrivere uno 0 logico nel corrispondente bit del registro di direzione; scrivendo un uno logico la linea viene configurata come uscita.

Registri di uscita ORB e ORA (Registri $0 e $1)

Se una linea è definita come uscita, un uno logico scritto nel corrispondente bit del registro di uscita provoca la generazione del livello alto sulla linea dati relativa; uno zero manda la linea dati a livello basso.
Leggendo dalla porta A una linea definita di uscita si ottiene il livello logico presente su quella linea. Leggendo dalla porta B una linea definita di uscita si ottiene il valore precedentemente impostato con l'operazione di uscita ad essa relativa.

Registri di ingresso IRB e IRA (Registri $0 e $1)

Se una linea è definita di ingresso, la CPU leggerà un uno logico se il livello ad essa applicato corrisponde al livello uno TTL; se la tensione applicata corrisponde al livello zero TTL verrà letto uno zero logico.

Se una linea di ingresso viene lasciata fluttuante renderà un uno logico.
Esempio: configurazione della porta A come ingresso e della porta B come uscita. La lettura effettuata con la porta A verrà emessa dalla porta B.

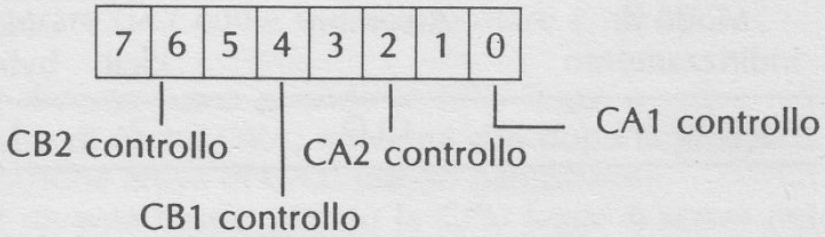
```
LDA ≠$00      ; carica 0000 0000 nel registro di direzione A
STA DDRA      ; tutte le linee della porta A sono di ingresso
LDA ≠$ FF     ; carica 1111 1111 nel registro di direzione B
STA DDRB      ; tutte le linee della porta B sono di uscita
LDA IRA       ; leggi i dati dalla porta A
STA ORB       ; ed emettili dalla porta B
```

Altro esempio: configurazione di PA7 e PA6 come ingresso e PA5...PA0 come uscita.

```
LDA ≠$3F      ; carica 0011 1111 nel registro di direzione A
STA DDRA      ; PA7-PA6 ingresso, PA5-PA0 uscita
```

Registro di controllo periferico (Registro $C)

Il registro di controllo periferico PCR seleziona il modo di operare delle linee di controllo CA1, CA2, CB1, CB2. È formato come indicato nella figura 3.



Linea di controllo CA1

CA1 può essere configurata solo come linea di ingresso; il bit 0 di PCR definisce a quale transizione di livello dovrà essere impostato a 1 il bit dell'indicatore degli interrupt (IFR).

0 transizione da alto a basso
1 transizione da basso a alto

L'indicatore può essere azzerato da un'operazione di lettura o scrittura del registro di uscita ORA.

Linea di controllo CA2

CA2 è una linea bidirezionale e viene configurata dai bit 1,2 e 3 del registro PCR.

— Per configurare CA2 come input impostare a 0 il bit 3
il bit 1 definisce il tipo di input:
0 = modo di ingresso normale
1 = modo interrupt
il bit 2 definisce la transizione attiva di CA2:
0 = transizione da alto a basso
1 = transizione da basso a alto

Una transizione attiva di CA2 imposta a uno il bit 0 del registro dei segnalatori di interrupt (IFR).
Nel modo di ingresso normale, l'indicatore di interrupt CA2 del registro IFR viene azzerato da una lettura o scrittura del registro di uscita ORA. Nel modo di ingresso interrupt l'indicatore di CA2 può essere azzerato da una scrittura nel registro degli indicatori di interrupt (vedi caratteristiche dell'interrupt).

— Per configurare CA2 come uscita impostare a 1 il bit 3

2	1	bit
0	0	CA2 diventa bassa quando la CPU legge e scrive nel registro di uscita ORA; ridiventa alto dopo la prossima transizione attiva di CA1. (modo handshake)
0	1	CA2 diventa basso quando la CPU legge o scrive nel registro di uscita ORA; ridiventa alto alla prossima transizione negativa del clock di sistema. (impulso)
1	0	CA2 diventa basso
1	1	CA2 diventa alto

Linee di controllo CB1 e CB2

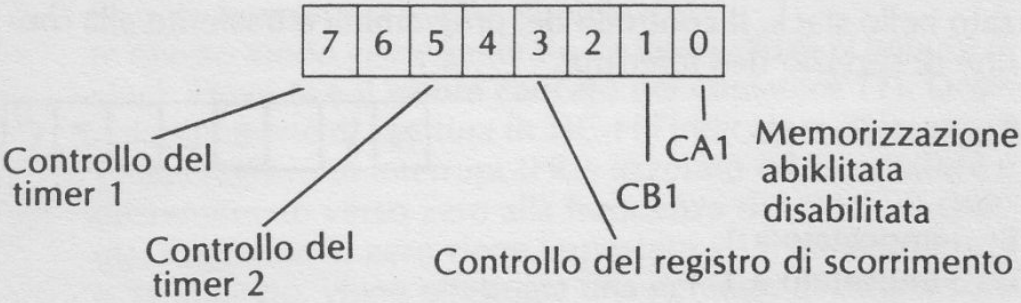
Le linee di controllo CB1 e CB2 vengono configurate dai restanti 4 bit di PCR. Il bit 4 configura CB1 ed i bits 5, 6, 7 configurano CB2

nello stesso modo in cui i bit 0 e 1, 2 e 3 configurano CA1 e CA2.
Ci sono però alcune piccole differenze:

A) CBI può essere usato come linea di uscita (cotrariamente a CA1) per l'uso con lo shift register
b) quando CB2 è impostato come uscita nei modi handshake o a impulso, diventerà basso solo per una scrittura nel registro di uscita ORA.

Registro di controllo ausiliario (registro $B)

Il registro di controllo ausiliario viene usato per configurare i due timer e lo shift register; inoltre abilita o disabilita la memorizzazione dei dati per le porte A e B. La funzione di ogni bit del registro ACR è mostrata nella figura 4.



Memorizzazione dei dati

I dati presenti alle porte A e B possono essere memorizzati al momento dell'ultima transizione attiva delle linee di controllo CA1 e CB1. Per impostare la memorizzazione i bit 0 e 1 del registro di controllo ausiliario (ACR) devono essere a uno mentre per disabilitare la memorizzazione devono essere a zero. La memorizzazione viene disabilitata non appena la CPU legge l'appropriato registro di input ORA o ORB.
Esempio: configurazione della VIA per abilitare la memorizzazione dei dati per la porta A e per disabilitare la memorizzazione per la porta B:

```
LDA ≠$01        ; combinazione 0000 0001 dei bit
```

STA ACR ; memorizzazione abilitata per la porta A e disabilitata per la porta B

Alla prossima transizione attiva di CA1 (definita dal bit 0 del registro di controllo periferico) i dati verranno memorizzati dalla porta A. La prossima lettura del registro ORA vedrà i dati memorizzati e non quelli presenti in quell'istante alle linee della porta A.

Timer 1

Il timer 1 è un contatore a 16 bit con due registri a 8 bit che permettono di memorizzare il byte meno significativo ed il byte più significativo. Sono disponibili 4 modi di funzionamento selezionabili con i bit 6 e 7 del registro di controllo ausiliario (ACR).

7 6 bits

0 0 Singolo interrupt con PB7 disabilitato.
In questo modo viene generato un intervallo di N impulsi di clock (dove N è il valore caricato nel contatore T1). Dopo un'operazione di scrittura in TIC-H l'indicatore di interrupt T1 del registro di interrupt IFR è azzerato ed il contatore è decrementato verso zero alla frequenza del clock di sistema. Raggiunto lo zero viene impostato a 1 il bit di interrupt e, se abilitato, viene effettuata una richiesta di IRQ. Il contatore continua a decrementare oltre lo zero se l'interrupt è stato abilitato. Il tempo trascorso dalla richiesta di interruzione può così essere determinato leggendo il contatore.

0 1 Interrupt multiplo con PB7 disabilitato
In questo modo viene generata una certa serie di interrupt con un periodo pari a N volte gli impulsi del clock di sistema, dove N è il numero caricato nel contatore T1.
La serie di interrupt inizia dopo un'operazione di scrittura in TIC-H che provoca anche l'azzeramento dell'indicatore di interrupt nel registro IFR. Il contatore viene quindi decrementato alla frequenza del clock di sistema fino al raggiungimento dello zero. Raggiunto lo zero viene impostato a uno l'indicatore di interrupt, il contenuto iniziale del contatore viene reimpostato ed il processo viene ripetuto.

1 0 Emissione di un impulso da PB7.
In questo modo viene generato un impulso negativo con la linea PB7 di durata pari a N impulsi di clock, dove N è il numero impostato in T1. Dopo la scrittura in T1C-H, PB7 diventa basso e rimane tale fino a che il contatore raggiunge lo zero; a questo punto PB7 ridiventa alto. Naturalmente PB7 deve essere stato precedentemente impostato come linea di uscita nel DDRB.

1 1 Emissione continua di impulsi da PB7.
In questo modo si genera una forma d'onda quadra in uscita da PB7. Dopo una scrittura in T1C-H, PB7 va bassa e ritorna alta quando il contatore viene decrementato a zero, impostando a uno l'indicatore di interrupt.
A questo punto viene ripristinato il contenuto del contatore, azzerato l'indicatore di interrupt e generato un impulso opposto al precedente di pari durata. Queste operazioni vengono ripetute continuamente.
NB: per calcolare la precisa durata degli impulsi, consultare il data sheet del costruttore.

Lettura e scrittura del contatore T1

Il contatore a 16 bit viene caricato con le seguenti istruzioni: il byte meno significativo è caricato nel registro di memoria meno significativo con una scrittura in T1C-L; il byte più significativo è caricato nel registro di memoria più significativo con una scrittura in T1C-H.
Dopo la scrittura in T1C-H, il contenuto di entrambi i registri di memoria è trasferito nel contatore T1; l'indicatore di interrupt per T1 è azzerato ed il contatore decrementato alla frequenza del clock di sistema.

Esempio: caricamento del contatore T1 con $8024.

```
LDA ≠$24      ;
STA T1C-L     ; carica il byte meno significativo
LDA ≠$80      ; carica il byte più significativo
```

```
STA T1C-H      ; viene caricato T1, Azzerato l'indicatore di in-
                 terrupt ed inizia il decremento del contatore alla
                 frequenza del clock di sistema
```

Il contatore a 16 bit può essere letto con le seguenti istruzioni: il contatore meno significativo viene letto da T1C-L, l'indicatore di interrupt di T1 viene azzerato, il contatore più significativo è letto da T1C-H.

Esempio: lettura del contenuto del contatore T1 e memorizzazione del suo contenuto nelle locazioni di memoria MEM e MEM+1

```
LDA T1C-L      ; il byte meno significativo viene letto
STA MEM        ; e memorizzato in MEM. L'indicatore è azze-
                 rato
LDA T1C-H      ; il byte più significativo viene letto
STA MEM+1      ; e memorizzato in MEM+1
```

Lettura e scrittura dei registri di memoria di T1

I registri di memoria del contatore T1 possono essere letti e scritti con le seguenti istruzioni:

```
LDA T1L-L      ; lettura dal registro meno significativo
STA T1L-L      ; scrittura nel registro meno significativo
LDA T1L-H      ; lettura del registro più significativo
STA T1L-H      ; scrittura nel registro più significativo
```

Non viene effettuato nessun trasferimento nel contatore e l'indicatore di interrupt resta invariato.

Timer 2

Il timer 2 è costituito da un contatore a 16 bit e da un registro per memorizzare il byte meno significativo; può avere due modi di

funzionamento, selezionabili mediante il bit 5 del registro ausiliario di controllo ACR.

Bit 5 Modo di funzionamento

0 Interrupt temporizzato
Dopo un intervallo di tempo pari a N impulsi del clock di sistema viene impostato l'indicatore di interrupt. Il conteggio (decremento) inizia dopo una scrittura in T2C-H.

1 Conteggio di inpulsi.
Questo modo di funzionamento usa il pin PB6, che deve essere configurato come ingresso, per contare impulsi.
Dopo la scrittura in T2C-H viene azzerato l'indicatore di interrupt ed il contenuto di T2 viene decrementato ad ogni impulso ricevuto da PB6. Quando T2 raggiunge lo zero viene impostato l'indicatore di interrupt.

Lettura e scrittura del contatore T2

Il contatore a 16 bit viene caricato con questa sequenza: il byte meno significativo è caricato nel registro T2C-L; il byte più significativo è caricato nel più significativo del contatore T2C-H. Dopo la scrittura in T2C-H il contenuto del registro meno significativo è copiato automaticamente nel byte meno significativo del contatore, l'indicatore di interrupt viene azzerato e il decremento inizia al prossimo impulso rilevato da PB6. La lettura del contenuto di T2 avviene leggendo prima T2C-L, che provoca l'azzeramento dell'indicatore di interrupt, e poi T2C-H.

Shift register (registro a scorrimento)

Il modo di funzionamento dello shift register è indicato dai bit 4, 3 e 2 del registro di controllo ausiliario ($B).
Sono possibili 7 modi di funzionamento oltre alla disabilitazione. L'ingresso o l'uscita seriale dei dati avviene attraverso la linea CB2 alla frequenza stabilita da un clock interno o da un clock esterno applicato alla linea CB1.

bit 4 3 2	Modo di funzionamento
0 0 0	Lo shift è disabilitato ed il controllo delle linee CB1 e CB2 avviene attraverso il registro PCR. L'indicatore di interrupt è azzerato.
0 0 1	La frequenza di trasferimento è data dal timer T2 e dalla frequenza del clock di sistema. Il trasferimento dati inizia dopo una lettura o scrittura dello shift register; l'indicatore di interrupt viene impostato dopo la ricezione di otto impulsi su CB1.
0 1 0	La frequenza di trasferimento è data dal clock di sistema. Il trasferimento inizia dopo una lettura o scrittura dello shift register; l'indicatore di interrupt viene impostato dopo la ricezione su CB1 di otto impulsi.
0 1 1	La frequenza di trasferimento è determinata da un dispositivo esterno. L'indicatore di interrupt viene impostato dopo otto impulsi su CB1.
1 0 0	I dati vengono emessi ad una frequenza controllata dal timer T2 e dal clock di sistema. Il trasferimento inizia dopo una lettura o scrittura dello shift register.
1 0 1	I dati vengono emessi ad una frequenza determinata da T2 e dal clock di sistema. Il trasferimento inizia dopo la lettura o la scrittura dello shift register. L'indicatore di interrupt viene impostato dopo otto impulsi di CB1.
1 1 0	I dati vengono emessi alla frequenza del clock di sistema. Il trasferimento inizia dopo una lettura o scrittura dello shift register. L'indicatore di interrupt viene impostato dopo otto impulsi.
1 1 1	I dati vengono emessi in base agli impulsi forniti da un dispositivo esterno. Il trasferimento inizia dopo una lettura o scrittura dello shift register. L'indicatore di interrupt viene impostato dopo otto impulsi.

Gestione degli interrupt

La VIA 6522 ha una sola linea di richiesta interrupt attiva bassa. Questa linea viene abilitata o disabilitata in base allo stato dei registri di interrupt IFR e abilitazione interrupt IER.

Registro degli indicatori di interrupt (registro $D)

La VIA ha un registro di indicatori di interrumpt con sette indicatori ed un bit IRQ che indica se la linea $\overline{IRQ}$ è abilitata o disabilitata; il formato del registro è indicato nella figura 5

Fig. 5

IRQ	T1	T2	CB1	CB2	SR	CA1	CA2

IRQ impostato a uno quando uno qualunque degli altri bit è a uno.

T1 impostato a uno quando il contatore T1 raggiunge lo zero. Viene azzerato da una lettura in T1C-L o da una scrittura in T1C-H

T2 impostato a uno quando il contatore T2 raggiunge lo zero. Viene azzerato da una lettura in T2C-L o da una scrittura in T2C-H.

CB1 impostato a uno dopo una transizione attiva di CB1. Viene azzerato da una lettura o scrittura in ORB.

CB2 impostato a uno dopo una transizione attiva di CB2. Viene azzerato da una lettura o scrittura in ORB.

SR impostato a uno dopo otto spostamenti. Azzerato da una lettura o scrittura nello shift register.

CA1 impostato a uno dopo una transizione attiva di CA1. Viene azzerato da una lettura o scrittura in ORA.

CA2 Impostato a uno dopo una transizione attiva di CA2. Viene azzerato da una lettura o scrittura in ORA.

N.B. i singoli bit possono essere azzerati scrivendo un uno nella corrispondente posizione di IFR.

Registro di abilitazione degli interrupt

Gli interrupt $\overline{IRQ}$ possono essere abilitati o disabilitati con la scrittura nel registro di abilitazione degli interrupt, il cui formato è simile al registro degli indicatori ad eccezione del bit 7 (vedi Fig. 6)

Fig. 6

7	6	5	4	3	2	1	0
S/C	T1	T2	CB1	CB2	SR	CA1	CA2

L'interrupt può essere disabilitato scrivendo uno zero nel bit 7 insieme ad un uno nel bit appropriato.

Esempio: LDA ≠$03 ; 0000 0011

STA IER ; disabilita CA1 e CA2

L'interrupt può essere abilitato scrivendo un uno nel bit 7 insieme ad un uno nel/nei bit appropriati.

Esempio: LDA ≠$92 ; 1001 0010

STA IER ; abilita gli interrupt per CB1 e CA1

Quando viene effettuata una lettura di IER il bit 7 appare sempre a zero, mentre gli altri bit sono a uno se abilitati, e a zero se il rispettivo interrupt è disabilitato.

Richesta di $\overline{IRQ}$

La richiesta di $\overline{IRQ}$ viene generata quando almeno un indicatore di IFR è alto ed il corrispondente bit di IER è abilitato. Quando il bit 7 di IFR è alto vuol dire che è stata emessa una richiesta di $\overline{IRQ}$.

Interrupt e reset

Il microprocessore 6502 ha due linee hardware di interrupt: NMI (interrupt non mascherabile) e $\overline{IRQ}$ (richiesta di interrupt). En-

trambe le linee sono attive basse ma $\overline{IRQ}$ è sensibile al livello mentre NMI è sensibile all'impulso.
Gli indirizzi per le due routine di interrupt sono memorizzati nelle seguenti locazioni di memoria.

FFFA	parte bassa del byte	$\overline{NMI}$
FFFB	parte alta del byte	
FFFC	parte bassa del byte	$\overline{RES}$
FFFD	parte alta del byte	
FFFE	parte bassa del byte	$\overline{IRQ}$ e BRK
FFFF	parte alta del byte	

L'indirizzo della routine di Reset verrà discussa nella sezione del reset; l'indirizzo della routine di servizio di BRK, che è un interrupt software, è lo stesso di quello di $\overline{IRQ}$.

Richiesta di interrupt (IRQ)

Questa linea viene normalmente collegata alla linea di richiesta interrupt dei dispositivi di immissione/emissione; nel caso della VIA sarà collegata a $\overline{IRQ}$. L'interrupt può essere disabilitato impostando il bit apposito nel registro di stato della CPU. (Vedi le istruzioni SEI e CLI). Quando la linea $\overline{IRQ}$ diventa bassa viene richiesto un interrupt e vengono eseguite le seguenti operazioni:

1) viene completata l'istruzione corrente
2) viene controllato il bit I del registro di stato, se risulta uguale a uno l'interrupt viene ignorato ed il programma prosegue con la prossima istruzione. L'indicatore di interrupt viene impostato e potrà quindi essere testato in seguito.
3) se l'indicatore I non era a uno la CPU salva nello stack il contatore di programma (prima il byte più significativo) ed il contenuto del registro di stato
4) Viene disabilitata la richiesta di IRQ per prevenire ulteriori interrupt e viene impostato l'indicatore di BREAK.
5) l'indirizzo della routine di servizio viene ottenuto dalle locazioni $FFFE e $FFFF e caricato nel contatore di programma

trasferendo così il controllo alla routine di servizio dell'interrupt.

Interrupt non mascherabile

Un impulso negativo su questa linea genera un'interruzione che non può essere disabilitata e che provoca l'esecuzione delle seguenti operazioni:

1) viene completata l'istruzione corrente
2) la CPU salva dello stack il contenuto del contatore di programma, prima il byte più significativo, e del registro di stato.
3) viene disabilitata la richiesta di IRQ ed impostato a uno l'indicatore di BREAK
4) l'indirizzo della routine di servizio viene ottenuta dalle locazione $FFFA e $FFFB e caricato nel contatore di programma trasformando così il controllo alla routine di gestione dell'interrupt.

NB: l'interrupt NMI ha una priorità più alta di IRQ.

Salvataggio dello stato della CPU

Il contenuto dell'accumulatore e dei registri indice non viene salvato automaticamente dall'ingresso nella routine di interrupt. Il contenuto di questi registri può essere salvato nello stack con le seguenti istruzioni.

```
PHA ; salva l'accumulatore
TYA ;
PHA ; salva il registro Y
TXA ;
PHA ; salva il registro X
```

I dati possono essere ripristinati prima dell'uscita dalla routine di interrupt con le istruzioni.

```
PLA ;
TAX ; ripristina il contenuto del registro X
PLA ;
TAY ; ripristina il contenuto del registro Y
PLA ; ripristina il contenuto dell'accumulatore
```

Ritorno da un interrupt

Le routine di servizio per $\overline{IRQ}$ e NMI terminano, normalmente, con una istruzione di RTI. Questa istruzione incrementa il puntatore dello stack di uno e ricarica il registro di stato con il byte così puntato; i successivi due incrementi dello stack pointer permettono di prelevare dallo stack il contenuto del contatore di programma che viene così ripristinato per proseguire il programma interrotto dall'interrupt.

Richieste multiple di $\overline{IRQ}$

La maggior parte dei dispositivi di immissione/emissione ha più di una richiesta di IRQ; la VIA 6522 ne ha, per esempio, ben 7. La sorgente dell'interrupt può essere determinata testando i singoli bit del registro di interrupt. L'ordine con cui i vari bit vengono controllati determina la priorità che il programmatore assegna ad ogni interrupt. Se più di un dispositivo può richiedere un interrupt alla CPU bisognerà provvedere a controllare tutti i dispositivi e si rallenterà quindi la risposta all'interrupt.

Reset

Questa linea è usata per inizializzare il microprocessore ad ogni accensione. Quando questa linea viene abbassata la CPU cessa ogni operazione di immissione/emissione e, quando la linea torna alta, viene eseguita la sequenza:

1) Viene impostato a uno il bit di disabilitazione dell'interrupt

2) L'indirizzo della routine di interrupt è prelevato dalle locazioni di memoria $FFFC e $FFFD e caricato nel contatore di programma, trasferendo così il controllo alla routine di gestione del reset.

Questa linea è spesso usata per riprendere il controllo del microprocessore quando, per errore di programma, si entra in un loop infinito.

BRK

L'istruzione di BREAK corrisponde ad un interrupt software e la sua esecuzione imposta a uno l'indicatore di BREAK nel registro di stato. La CPU salva il contenuto del contatore di programma (prima il byte più significativo) nello stack, salva pure nello stack il registro di stato e imposta a uno il bit di disabilitazione delle richieste di IRQ per disabilitarle. L'indirizzo della routine di gestione viene ricavato dalle locazioni $FFFE e $FFFF e caricato nel contatore di programma trasferendo così il controllo alla routine di servizio.

Per distinguere una richiesta di IRQ da un BRK bisogna pelevare dallo stack il registro di stato e controllare l'indicatore di break.

ESEMPI

Somma binaria a 8 bit

I due numeri da sommare sono memorizzati nelle locazioni indicate con MEM e MEM+1, il risultato viene depositato nelle locazioni ANSWRL e ANSWRH.

```
CLD          ; aritmetica binaria
CLC          azzera il riporto
LDA #$00     ;
STA ANSWRH   ; azzera il byte più si-        MEM
             gnificativo del risulta-
             to
ADC MEM      ; primo numero                  MEM+1
```

```
ADC MEM+1      ; secondo numero
STA ANSWRL     ; memorizza risultato
BCC CONT       ; se non c'è riporto
               OK
INC ANSWRH     in caso di riporto in-
               crementa il byte più
               significativo
CONT -----     ; continuazione del
               programma
```

ANSWRL
ANSWRH

NB:

1) le istruzioni del 6502 non comprendono la somma senza riporto, perciò il riporto dovrà essere azzerato prima di iniziare una somma.
2) L'indicatore dell'aritmetica binaria deve essere azzerato per essere sicuri che il calcolo non venga eseguito in BCD

Sottrazione binaria a 8 bit.

Il minuendo è memorizzato nella locazione MEM ed il sottraendo nella locazione MEM+1; il risultato verrà memorizzato nelle locazioni ANSWRL e ANSWRH.

```
CLD            ; aritmetica binaria
SEC            ; imposta a uno il ri-
               porto
LDA #$00       ;
STA ANSWRH     ; azzera il byte più si-
               gnificativo del risulta-
               to
LDA MEM        ; carica il minuendo
SBC MEM+1      ; sottrai il sottraendo
STA ANSWRL     ; memorizza il byte
               più significativo
BCS CONT       ; se il riporto è a uno
               continua con il pro-
               gramma principale
```

MEM

MEM+1

ANSWRL

ANSWRH

```
DEC ANSWRH   ; decrementa di uno il
             byte più significativo
CONT -----   ; continua con pro-
             gramma principale
```

NB:

a) L'indicatore di riporto deve essere impostato a uno prima della sottrazione. Quando il sottraendo è più grande del minuendo l'indicatore verrà azzerato.

b) L'indicatore dell'aritmetica binaria deve essere azzerato per essere sicuri che il calcolo non venga eseguito in BCD.

Somma binaria a 16 bit

I due numeri a 16 bit da sommare sono memorizzati rispettivamente nelle locazioni NUMB1H, NUMB1L e NUMB2H, NUMB2L; il risultato verrà memorizzato in ANSWRL, ANSWRM e ANSWRH.

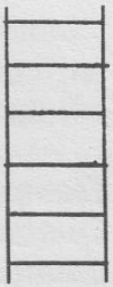
```
CLD          ; aritmetica binaria
CLC          ; azzera il riporto
LDA #$00     ;                          NUMB1L
STA ANSWRH   ; azzera il byte più si-   NUMB1H
             gnificativo
ADC NUMB1L   ;
ADC NUMB2L   ; somma il byte meno       NUMB2L
             significativo
STA ANSWRL   ; memorizza la parte       NUMB2H
             meno significativa del
             risultato
LDA #$00     ; azzera l'accumula-
             tore
ADC NUMB1H   ;
             ;                          ANSWRL
ADC NUMB2H   ; somma il byte più si-    ANSWRM
             gnificativo del primo      ANSWRH
             addendo
```

```
STA ANSWRM     ; memorizza il risultato intermedio
BCC CONT       ; se il riporto è zero salta al pgm principale
INC ANSWRH     ; incrementa il byte più significativo del secondo addendo
CONT -----     ; continua con il pgm principale
```

Moltiplicazione a 8 bit

La moltiplicazione verrà effettuata con la tecnica dello "spostamento con somma". Il moltiplicatore è memorizzato nella locazione MULP ed il moltiplicando in MULT. Il risultato sarà memorizzato in ANSWRL e ANSWRH. COUNT contiene il numero di shift (spostamenti) da effettuare

```
CLD            ; aritmetica binaria
LDA #$00       ; azzera alcune locazioni di memoria
STA MULTH      ;
STA ANSWRL     ;
STA ANSWRH
LDA #$08       ;
STA COUNT      ; imposta COUNT a 8
AGAING ROR     ; esamina il primo bit
MULP
BCC SHIFT      ; se zero non effettuare la somma
CLC            ; azzera il riporto
LDA MULTL      ;
ADC ANSWRL     ; somma il byte meno significativo
```



```
 STA ANSWRL     ; e memorizza il risul-
                  tato
 LDA MULTH      ;
 ADC ANSWRH
 STA ANSWRH     ; somma il byte più si-
                  gnificativo del risulta-
                  to
SHIFT           ; e memorizza
ASLMULTL
 ROL MULTH      ; ruota di un bit il mol-
                  tiplicando
 DEC COUNT      ; decrementa il con-
                  tatore
 BNE AGAIN      ; se diverso da zero
                  va ad esaminare il pri-
                  mo bit
CONT -----
```

COUNT

Gestione di un IRQ

Inizializzazione
La porta 8 della VIA viene configurata come uscita e la porta A come ingresso; viene abilitata la richiesta di interrupt per la linea CAI e disabilitate tutte le altre.

```
SEI          ; disabilita IRQ
LDX #$FF     ;
TXS          ; imposta lo stack pointer
LDA #$00     ; azzera l'accumulatore
STA PCR      ; la transizione attiva per CA1 è 1→ 0
STA DDRA     ; la porta A è configurata per l'ingresso
LDA #$FF     ;
STA DDRB     ; la porta B è configurata per l'uscita
LDA #$7F     ;
STA IER      ; disabilitazione degli interrupts
LDA #82      ;
```

```
STA IER          ; abilita l'interrupt per CA1
CLI              ; abilita IRQ
-----            ; segui con il pgm principale
```

Routine di servizio dell'interrupt

```
LDA IRA          ; azzera l'indicatore di interrupt per CA1
....             ; continua con la routine di interrupt
RT1              ; fine della routine di interrupt
```

NB:

a) lo stack viene inizializzato

b) IRQ viene disabilitato durante l'inizializzazione per evitare che una richiesta la interrompa pregiudicando la corretta ripartenza. Verrà riabilitato alla fine.

c) La prima istruzione della routine di servizio potrebbe essere rimpiazzata da STA ORA per azzerare l'indicatore di CA1

d) Le locazioni $FFFE e $FFFF devono contenere, direttamente o indirettamente, l'indirizzo della routine di gestione dell'interrupt.

e) per distinguere una richiesta di IRQ da un BRK sarà necessario includere nella routine di servizio il test del bit di BRK del registro di stato salvato nello stack.

Generazione di un'onda quadra su PB7 con il timer 1

```
LDA #$7F         ; maschera binaria 0111 1111;
STA IER          ; disabilita tutti gli interrupt
LDA #$FF         ; maschera binaria 1111 1111
STA DDRB         ; la porta B viene configurata per emissione
LDA #$C0         ; maschera 1100 0000
STA ACR          ; viene selezionato il modo 11 del timer 1 (uscita su PB7)
LDA #$mm         ;
STA T1C-L        ; carica il registro meno significativo del timer 1
```

```
LDA #$nn        ; reset dell'indicatore di interrupt ed inizio
                  decremento
STA T1C-H
CONT -----      ; ... continua
```

NB:

a) dopo la scrittura in T1C-H, la linea di uscita PB7 divente bassa, ritornerà alta quando il contatore raggiungerà lo zero, p. es. dopo $nnmm impulsi del clock di sistema; i registri verranno quindi automaticamente ricaricati nel contatore ed il processo si ripeterà.

b) modificando il contenuto di ACR a ≠$80 invece di $≠00, verrebbe selezionato il modo 01 che permette l'emissione di un singolo impulso negativo su PB7.

Conteggio di impulsi con il timer 2

Configurazione del timer 2 per contare un certo numero di impulsi applicati a PB6; l'interrupt T2 è disabilitato.

```
LDA #$7F        ; maschera binaria 0111 1111
STA IER         ; disabilita tutti gli interrupt
LDA #$BF        ; maschera binaria 1011 1111
STA DDRB        ; la linea 6 della porta B è un ingresso, le
                  altre son uscite
LDA #$20        ; maschera binaria 0010 0000
STA ACR         ; selezione del modo 1 con il bit 5 di ACR
                  alto
LDA #$mm        ; caricamento registro meno significativo
                  del timer
STA T2C-L       ;
LDA #$nn        ; al caricamento del byte più significativo
                  viene automaticamente trasferito
STA T2C-H       ; il contenuto del registro meno significa-
                  tivo nel contatore ed azzerato l'indicatore
                  di interrupt
```

```
LOOP LDA IFR      ; leggi il registro degli interrupt
 AND #$20         ; isola il bit 5
 BEQ LOOP         ; se zero aspetta e ripeti il ciclo
```

NB:

a) la linea PB6 deve essere configurata per l'immissione
b) quando ≠ $nnmm saranno ricevuti da PB6, sarà impostato a uno l'indicatore di interrupt
c) la richiesta di IRQ è disabilitata

Generazione di una forma d'onda con CB2

La frequenza generata da CB2 è di 500 Hz con un rapporto impulso/pausa di 1:1

```
AGAIN LDA #$EO    ; maschera binaria 1110 0000
 STA PCR          ; impulso a 1 CB2
 JSR DELAY        ; routine di ritardo
 LDA #$C0         ; maschera binaria 1100 0000
 STA PCR          ; imposta a 0 CB2
 JSR DELAY        ; routine di ritardo
 JMP AGAIN        ; ripeti il ciclo
DELAY ...         ; routine di ritardo di 1ms
RTS
```

Conversione analogica/digitale

Alla VIA è collegato un convertitore A/D a 10 bit. Esso richiede un impulso di conversione (transizione 1 → 0 → 1) e abbassa la sua linea di stato quando ha finito la conversione. Viene usato CA1 per controllare lo stato del convertitore e CA2 per generare l'impulso di inizio conversione. Il risultato a 10 bit viene memorizzato nelle locazioni MEM e MEM+1. Vedi la fig. 7.

```
LDA #$00          ; azzera l'accumulatore
 STA DDRA         ; configura la porta A come ingresso
 STA DDRB         ; configura la porta B come ingresso
```

```
STA PCR        ; transizione attiva su CA1 1 → 0
LDA #$0E       ;
STA PCR        ; maschera 0000 1110 imposta a 1 CA2
LDA #$0C
STA PCR        ; maschera binaria 0000 1100 azzera
               CA2
LDA #$0E
STA PCR        ; maschera binaria 0000 1110 imposta a 1
               CA2
WAIT LDA IFR   ; leggi il registro degli indicatori
AND #$02       ; isola l'indicatore di CA1
BNE WAIT       ; fine conversione?
LDA IRA        leggi i bits meno significativi dalla porta
               A
STA MEM+1      ; memorizzati in MEM+1
LDA IRB        ; leggi la porta B
AND #$03       ; mantieni solo i primi due bits
STA MEM        ; e memorizzati in MEM
```

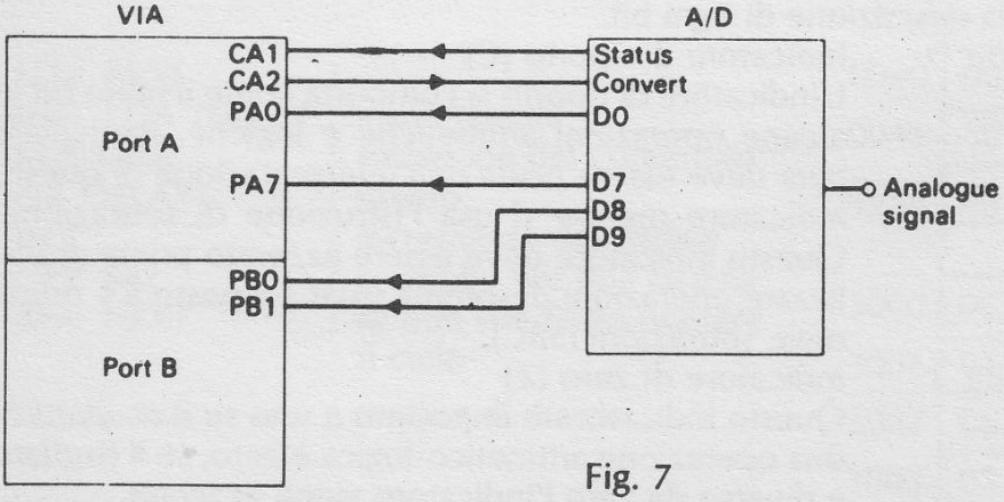


Fig. 7

Configurazione di più VIA

Configurazione della porta A di alcune VIA usando l'indirizzamento indirizzato indiretto. La tabella di memoria in pagina zero contiene gli indirizzi dei registri di direzione dati.

```
LDX #$00          ;
LDA #$0FF         ; combinazione
                    1111 1111 per linee
                    di uscita
STA ($80,X)       ; DDRA1 configurato
STA ($82,X)       ; DDRA2 configurato
STA ($84,X)       ; DDRA3 configurato
STA ($86,X)       ; DDRA4 configurato
```

0080	DDRA1l
0081	DDRA1h
0082	DDRA2l
0083	DDRA2h
0084	DDRA3l
0085	DDRA3h
0086	DDRA4l
0087	DDRA4h

Emissione dalla porta B

Emissione tramite la porta B di una tabella usando l'indirizzamento indiretto indicizzato.

```
 LDY #$00          ;
NEXT LDA ($40),Y   ; inizio tabella
 STA ORA           ; emissione del contenuto
 jSR DELAY         ; ritardo
 INY               ; incremento dell'indice
 CPY $42           ; controllo per fine tabella
 BNE NEXT          ; se non si è finito ripetere
                     il ciclo

DELAY -----        ; routine di ritardo
 RTS
```

0040	00
0041	30
0042	08

3000	0C
3001	04
3002	06
3003	02
3004	03
3005	01
3006	09
3007	08

NB:

a) L'indirizzo della tabella è memorizzato nelle locazioni $40 e $41 di pagina zero
b) il numero di elementi della tabella è memorizzato nella locazione $42
c) i dati sono memorizzati nella tabella $3000-$3007

Cambiando l'indirizzo in pagina zero si può accedere ad un'altra tabella senza variare il programma.

Controllo di interruttori

Alla porta A della VIA sono collegati 8 interruttori; controllo del loro stato con le istruzioni logiche. (interruttori SW0-SW7)

— primo metodo: uso dell'AND per controllare SW1

```
LDA IRA        ; leggi tutti gli interruttori
AND ≠$02       ; l'accumulatore conterrà 0000 00X0, usare
                 BNE o BEQ per controllare l'indicatore di
                 zero
```

— secondo metodo. uso dell'istruzione ROR per l'interruttore SW0

```
LDA IRA        ; leggi tutti gli interruttori
ROR A          ; lo stato di SW0 va nell'indicatore di riporto,
                 usare BCC o BCS per controllarne lo stato
```

— terzo metodo: uso di BPL e BMI per l'interruttore SW7

```
LDA IRA        ; l'indicatore N è impostato come il bit 7, usare
                 BPL o BMI per controllarne lo stato
```

— quarto metodo: uso di BIT per controllare i bit 7, 6 e n

```
LDA ≠$08       ; maschera 0000 1000 per il bit 3
BIT IRA        ; and logico con l'accumulatore, l'indicatore 2
                 può essere usato per testare il bit 3. N e V sono
                 invece caricati con i bits 7 e 6
```

Può essere usata una combinazione di queste tecniche per adeguarsi alla situazione particolare.

Note

Note

Note

Altri libri di questa collana Jackson:

Computer

Sono manuali d'uso di immediata e semplice consultazione, da tenere a portata di mano di fianco al computer.

- **Sinclair Spectrum • VIC 20 • Commodore 64 • PC IBM**
- **Apple IIc - IIe • Sharp MZ80A**

Informatica

Forniscono le conoscenze fondamentali per una cultura informatica di base.

- **La programmazione**

Software

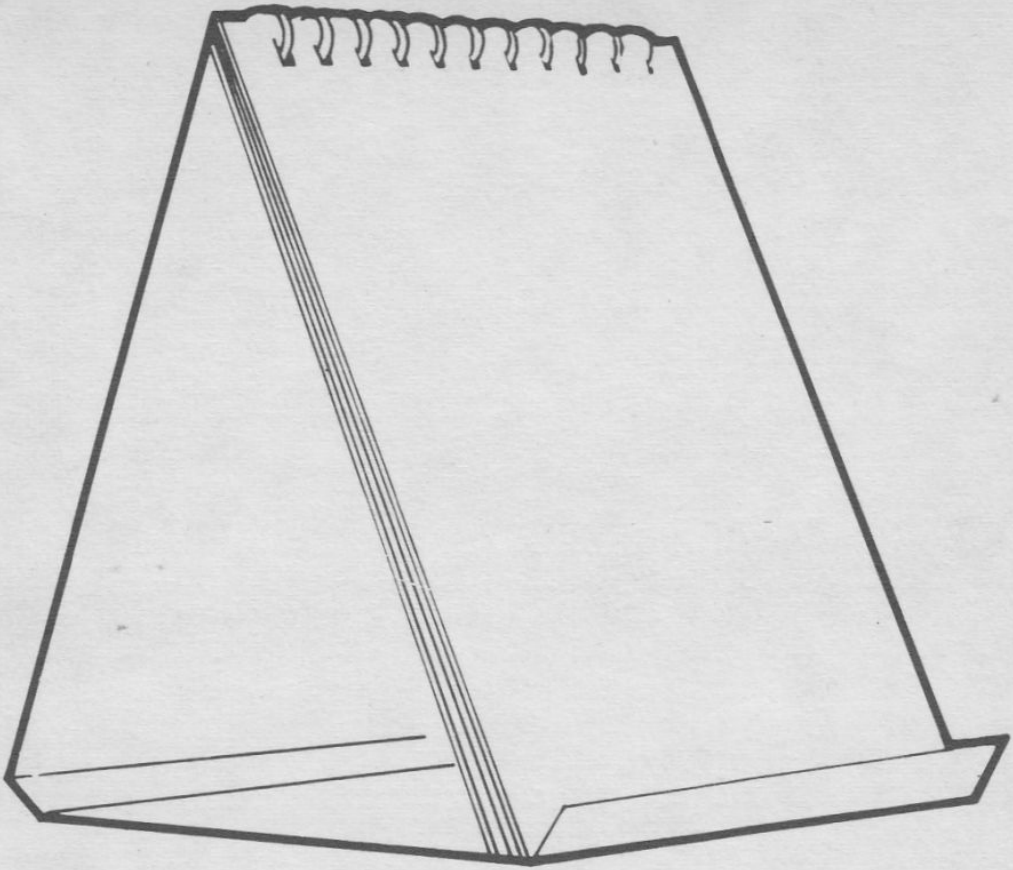
Presentano i pacchetti software e i sistemi operativi più diffusi sul mercato e suggeriscono idee e proposte di programmi per diverse applicazioni.

- **WordStar • UNIX • LOGO • MS-DOS • Programmi di statistica • CP/M • PC-DOS**

Linguaggi

Sono comode tabelle di riferimento delle istruzioni e i comandi dei linguaggi più famosi.

- **COBOL • FORTRAN 77 • PASCAL • BASIC**
- **Assembler Z80 • Assembler 6502**



L'illustrazione mostra come
sistemare il tascabile
per una più agevole consultazione.

I tascabili Jackson sono uno strumento prezioso per chi lavora con il computer. In poche pagine riassumono con chiarezza quanto è necessario sapere su diversi argomenti di informatica, andando incontro alle più diverse esigenze.

COD 013H ISBN 88-7056-243-3 **L 8.500**